

HOCHSCHULE MITTWEIDA – UNIVERSITY OF APPLIED SCIENCE (FH)
FACHBEREICH: MEDIEN

Die Nutzung der Blogsoftware WordPress als Web Content Management System

EINGEREICHT ALS: BACHELORARBEIT

1. Prüfer: Prof. Dr.-Ing. Robert J. Wierzbicki
2. Prüfer: Dipl.-Ing. Sieglinde Klimant

Maik Irmscher | Chemnitz, 2010

Bibliografische Beschreibung

Maik Irmscher

Die Nutzung der Blogsoftware `WordPress` als Web Content Management System. 2010.
124 Seiten. Mittweida, Hochschule Mittweida (FH) – University of Applied Sciences,
Fachbereich Medien, Bachelorarbeit, 2010.

Referat

Diese Bachelorarbeit beschäftigt sich mit der Nutzbarkeit der Weblogsoftware `WordPress` als Web Content Management System. Das Ziel dieser Abschlussarbeit ist es, einen Vergleich zwischen standardisierten Open Source Weblogsystemen und Open Source Web Content Management Systemen herauszuarbeiten. Außerdem soll die Arbeit als praktischer Leitfaden für Webdesigner- und entwickler dienen, um einen Überblick über die Programmlogik beziehungsweise den Aufbau von `WordPress` als Web Content Management System zu erhalten. Viele praktische Beispiele verdeutlichen die Strukturen des Systems und spiegeln den Einsatz in der Praxis wider. Des Weiteren werden explizit die Benutzeroberfläche von `WordPress`, sowie allgemeine Webentwicklungen analysiert. Im Resümee werden diese Untersuchung ausgewertet und zukünftige Richtlinien aufgezeigt.

Inhaltsverzeichnis

I. Abbildungsverzeichnis	7
II. Quelltextverzeichnis (Listings)	8
III. Abkürzungsverzeichnis	9
1. Einleitung	10
1.1 Ziel der Arbeit	10
1.2 Aufbau der Arbeit	11
1.3 Methodik und wissenschaftliches Vorgehen	11
2. Weblogs und Content Management Systeme	12
2.1 Weblogs	12
2.1.1 Allgemein	12
2.1.2 Technik und Software	14
2.1.3 Blogstudie	15
2.1.4 Vorteile von Weblogs	16
2.2 Content Management System	17
2.2.1 Allgemein	17
2.2.2 Technik und Software	19
2.2.3 Anforderung an Web Content Management Systeme	19
2.2.4 Zusätzliche Systemfeatures von Web Content Management Systemen	22
2.2.5 WCMS-Studie	22
2.3 Zusammenfassung Weblog und Content Management System	25

3. Einführung WordPress	26
3.1 Historische Entwicklung des Begriffs Weblog	26
3.2 Historische Entwicklung der Blogsoftware WordPress	28
3.2.1 WordPress Versionen im Überblick	30
4. WP-Backend und WCMS-Eigenschaften	34
4.1 Installation und Konfiguration	34
4.2 Administrationsoberfläche von WordPress	38
4.2.1 DASHBOARD	38
4.2.2 ARTIKEL	42
4.2.2.1 ARTIKEL ERSTELLEN und BEARBEITEN	43
4.2.2.2 TAGS und KATEGORIEN innerhalb der ARTIKEL	51
4.2.3 MEDIATHEK	51
4.2.4 LINKS	52
4.2.4.1 LINKS BEARBEITEN und LINKS HINZUFÜGEN	52
4.2.5 SEITEN	54
4.2.6 KOMMENTARE	55
4.2.7 DESIGN	55
4.2.8 PLUGINS	57
4.2.9 BENUTZER	59
4.2.9.1 Hinzufügen von AUTOREN und BENUTZERN	61
4.2.10 WERKZEUGE	61
4.2.11 EINSTELLUNG	61
4.2.11.1 ALLGEMEIN	62
4.2.11.2 SCHREIBEN und LESEN	62
4.2.11.3 DISKUSSIONEN, MEDIATHEK und PRIVATSPHÄRE	63
4.2.11.4 PERMALINKS	63

5. THEMES und TEMPLATES	64
5.1 Aufbau der Datei ,index.php' des DEFAULT-THEMES	64
5.1.1 Der LOOP	68
5.1.2 TEMPLATE TAGS	69
5.1.2.1 TEMPLATE TAGS ohne Parameter	70
5.1.2.2 TEMPLATE TAGS mit ,function-style' Parameter	70
5.1.2.3 TAGS mit ,query-string-style' Parameter	71
5.1.3 CONDITIONAL TAGS	71
5.1.3.1 Verwendung von CONDITIONAL TAGS in der Datei ,sidebar.php'	73
5.2 Aufbau der Datei ,header.php' des DEFAULT-THEMES	74
5.3 Hooks	76
5.3.1 ACTION HOOKS	76
5.3.2 FILTER HOOKS	79
6. Praktischer Einsatz von WordPress als WCMS	80
6.1 Anpassung von TEMPLATES	80
6.1.1 Einbindung von sIFR	81
6.1.2 SEITEN	83
6.1.3 Einbindung des Javascript-Frameworks JQUERY	84
6.2 Praxisnahe WCMS-Erweiterungen für WordPress	86
6.2.1 NEXTGEN GALLERY	86
6.2.2 Formulare mit CFORMS II	88
6.2.3 WP-GUESTBOOK PLUGIN	89
6.2.4 Suchmaschinenoptimierung mit WP-ALL IN ONE SEO	90
6.2.5 Weitere praxisrelevante WCMS-Erweiterung für WordPress	92
6.3 Nachteile von WordPress als WCMS	93

7. Vergleich zwischen TYPOLIGHT und WordPress	95
7.1 Kurze Einführung zum Open Source WCMS TYPOLIGHT	95
7.2 Ausgewählte WCMS-Eigenschaften im Vergleich	95
7.2.1 Installation	96
7.2.2 Administrationsoberflächen	97
7.2.3 Vorteile von TYPOLIGHT-Inhaltselemente gegenüber WordPress-Seiten	101
7.2.4 Vorteile von TYPOLIGHT-Modulen gegenüber WordPress-Widgets	102
7.2.5 Sortierung und Filterung von Datensätzen im Backend	103
7.2.6 Mehrfachbearbeitung	103
7.3 Dateiverwaltung	104
7.4 Rechteverwaltung	105
8. Resümee	107
IV. Anhang	109
CONDITIONAL TAGS	109
V. Literaturverzeichnis	111
VI. Erklärung zur selbstständigen Anfertigung	124

I. Abbildungsverzeichnis

Abbildung 2.1: Ergebnisse der WCMS-Studie, eigene Darstellung	24
Abbildung 3.1: Nutzerentwicklung von MOVABLE TYPE und WORDPRESS	29
Abbildung 4.1: WORDPRESS Installationsoberfläche	36
Abbildung 4.2: Passwortvergabe innerhalb des Installationsvorgang.	37
Abbildung 4.3: Login-Bereich von WORDPRESS mit dem Hinweis auf falschen Benutzername	38
Abbildung 4.4: DASHBOARD von WORDPRESS	39
Abbildung 4.5: Menüpunkt AKTUELL im DASHBOARD	40
Abbildung 4.6: Menüpunkt QUICKPRESS im DASHBOARD	40
Abbildung 4.7: LETZTE KOMMENTARE und AKTUELLE ENTWÜRFE	41
Abbildung 4.8: Konfiguration des FEEDREADER mit RSS-FEED von tagesschau.de	42
Abbildung 4.9: Artikelübersicht von WORDPRESS	43
Abbildung 4.10: Erstellung von ARTIKELN innerhalb des WORDPRESS-Backends	44
Abbildung 4.11: Einfügen von Medieninhalten	51
Abbildung 4.12: BILD HINZUFÜGEN in der MEDIATHEK	52
Abbildung 4.13: LINKS HINZUFÜGEN	53
Abbildung 4.14: Bedienfeld ATTRIBUTE im Menüpunkt SEITE ERSTELLEN.	54
Abbildung 4.15: PLUGINVERWALTUNG	58
Abbildung 6.1: Nutzung des PLUGINS JQUERY TABS AND SLIDES	85
Abbildung 6.2: Erstellung von Galerien mit der Erweiterung NEXTGEN GALLERY	86
Abbildung 6.3: Einfügen einer Galerie in SEITEN	87
Abbildung 6.4: Ausschnitt aus der Formular-Erweiterung CFORMS II	88
Abbildung 6.5: PLUGIN Einstellung für ALL IN ONE SEO	91
Abbildung 6.6: Vergabe von Schlüsselwörter in SEITEN mit Hilfe von ALL IN ONE SEO.	91
Abbildung 7.1: INSTALLTOOL VON TYPOLIGHT	96
Abbildung 7.2: DASHBOARD von WORDPRESS	97
Abbildung 7.3: DASHBOARD von TYPOLIGHT	98
Abbildung 7.4: Systemarchitektur von TYPOLIGHT, eigene Darstellung	100
Abbildung 7.5: Systemarchitektur von WORDPRESS, eigene Darstellung	100
Abbildung 7.6: TYPOLIGHT Inhaltselemente	101
Abbildung 7.7: Modulverwaltung von TYPOLIGHT	102
Abbildung 7.8: Auf- und Einklappen von SEITEN in TYPOLIGHT.	103
Abbildung 7.9: DATEIVERWALTUNG von TYPOLIGHT sowie MEDIATHEK von WORDPRESS	104
Abbildung 7.10: Einbinden von Bildern in TYPOLIGHT und WORDPRESS	105

II. Quelltextverzeichnis (Listings)

Listing 4.1: Ausschnitt der Konfigurationsdatei ‚wp-config.php‘	35
Listing 4.2: Zusatz: Mit ‚automatischer Speicherung‘ in der Datei ‚wp-config.php‘	35
Listing 4.3: Zusatz: Ohne ‚automatischer Speicherung‘ in der Datei ‚wp-config.php‘	36
Listing 4.3: Der TITEL im WORDPRESS DEFAULT-THEME	45
Listing 4.5: Verwendung von BENUTZERDEFINIERTEN FELDERN	48
Listing 4.6: Verwendung des TEMPLATE TAGS ‚the_date‘	50
Listing 4.7: Registrierung der SIDEBAR in der Datei ‚function.php‘ des THEMES	57
Listing 4.8: Implementierung der SIDEBAR in der Datei ‚sidebar.php‘ des THEMES	57
Listing 4.9: TEMPLATE TAG ‚bloginfo‘	62
Listing 5.1: Gesamtauszug der Datei ‚index.php‘ des DEFAULT-THEMES	65
Listing 5.2: Beginn des LOOPS	66
Listing 5.3: Ende des LOOPS	66
Listing 5.4: Die Gesamtdarstellung des LOOPS innerhalb der ‚index.php‘ des DEFAULT-THEMES	67
Listing 5.5: Darstellung eines ARTIKELS innerhalb des LOOPS	68
Listing 5.6: Fontend-Auflistung und Darstellung der einzelnen Beiträge	69
Listing 5.7: Beispiel für die Auflistung mit ‚query-string-style‘ Parameter	71
Listing 5.8: Fallunterscheidung mit Hilfe von CONDITIONAL TAGS	72
Listing 5.9: Ausschnitt aus der Datei ‚sidebar.php‘ des DEFAULT-THEMES	73
Listing 5.10: Ausschnitt aus der Datei ‚header.php‘ des DEFAULT-THEMES	75
Listing 5.11: PLUGIN HELLO DOLLY von MATT MULLENWEG	77
Listing 5.12: Ausschnitt aus dem PLUGIN HELLO DOLLY	78
Listing 5.13: Individuelle Anpassung des PLUGINS HELLO DOLLY	78
Listing 5.14: Einsatz von FILTER HOOKS	79
Listing 6.1: Individuelle Anpassung der Datei ‚index.php‘ des THEMES	80
Listing 6.2: Implementierung von sIFR in der Datei ‚functions.php‘	82
Listing 6.3: Verwendung des TEMPLATE TAG ‚wp_list_pages‘	83
Listing 6.4: Integration von Unterseiten innerhalb des THEMES	84
Listing 6.5: Einbindung von JQUERY aus dem WORDPRESS CORE	85
Listing 6.6: Auszug aus der Datei: ‚wp-guestbook_page.php‘	89
Listing 6.7: Einbindung einer Paginierung in der Datei ‚wp-paged-guestbook.php‘	90

III. Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
AGPL	Affero General Public License
API	application programming interface
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CMS	Content Management System
CSS	Cascading Stylesheets
CRM	Customer-Relationship-Management
DB	Datenbank
ERP	Enterprise Resource Planning
FTP	File Transport Protocol
GIF	Graphics Interchange Format
GNU	General Public License
HTML	Hypertext Markup Language
JPG	Joint Photographic Experts Group
LGPL	Lesser General Public License
MS	Microsoft
OS	operating system
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PNG	Portable Network Graphics
RSS	Really Simple Syndication
SCM	Supply-Chain-Management
SQL	Structured Query Language
sIFR	Scalable Inman Flash Replacement
swf	Shockwave Flash
URL	Uniform Resource Locator
UTF-8	8-bit UCS Transformation Format
TinyMCE	Tiny Moxiecode Content Editor
W3C	World-Wide-Web-Consortium
WCMS	Web Content Management System
WP	WordPress
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language
XML-RPC	Extensible Markup Language Remote Procedure Call

1. Einleitung

„WordPress ist weit mehr als eine Blog-Software. Dies beweist die stetig wachsende Zahl komplexer Webauftritte auf der Grundlage dieses Systems. Die große Entwicklergemeinschaft und der konsequent modulare Aufbau machen WordPress [...] zu einer ernsthaften Alternative, wenn die Programmierung von Webseiten mit HTML/PHP zu aufwändig, ein „großes“ Content Management System jedoch überdimensioniert ist.“ Frank Bülgte, 2009

Immer häufiger gewinnt unabhängige Publikation medialer Inhalte im Internet an Zuspruch. Dabei sollen vor allem technische Barrieren und der Einsatz externer Webdienstleister, sowie Fachpersonal aus Kostengründen vermieden werden. Gerade klein- und mittelständige Unternehmen, als auch Privatpersonen wollen ihre Webinhalte „selbst pflegen und verwalten“. Um einen Kompromiss zwischen verfügbarem Budget und den gewünschten Leistungen zu finden, greifen viele Webdesigner auf Open Source Web Content Management Systeme bei der Realisierung zurück. Da jedoch dieser Markt eine große Anzahl solcher Systeme anbietet, stellt sich für den Entwickler schnell die Frage: Welches ist das passende System?

Vor allem für kleine und mittlere Webauftritte sind eine Vielzahl dieser Systeme zu groß oder zu stattlich aufgebaut und verfehlen im alltäglichen Gebrauch das Ziel des Kunden. Ebenfalls benötigen viele dieser Systeme eine bestimmte Einarbeitungszeit. Warum also kein System nutzen, welches im Kern übersichtlich ist, sich dafür durch spezielle Erweiterung zielorientiert gestalten lässt?

In den vergangenen Monaten liest man in Fachblogs- und foren immer häufiger vom Einsatz der Blogsoftware WordPress als Web Content Management System, da WordPress die Basisvoraussetzung für die Erstellung dynamischer Webseiten beinhaltet. Des Weiteren existiert innerhalb des Systems eine übersichtliche und benutzerfreundliche Administrationsoberfläche, sowie die Möglichkeit, statische Seiten beziehungsweise Nachrichtenartikel zu erstellen. WordPress versucht sich in diesem Zusammenhang mit in die Reihe der zahllosen Open Source Web Content Management Systemen einzureihen, was zu beweisen ist.

1.1 Ziel der Arbeit

Das Ziel dieser Abschlussarbeit ist es, einen Vergleich zwischen standardisierten Open Source Weblogsystemen und Open Source Web Content Management Systemen herauszuarbeiten. Dabei liegt das Motiv vorrangig auf der Nutzbarkeit der Weblogsoftware WordPress unter Anwendungsaspekten eines Web Content Management Systems. Somit werden die Eigenschaften

eines Web Content Management Systems nachgewiesen, sofern diese existieren. Außerdem soll die Arbeit als praktischer Leitfaden für Webdesigner- und entwickler dienen, um einen Überblick über die Programmlogik beziehungsweise den Aufbau von WordPress als Web Content Management System zu erhalten. Des Weiteren werden Erweiterungsmöglichkeiten von WordPress gesucht und unter praktischen Erfahrungen dargestellt.

1.2 Aufbau der Arbeit

Zu Beginn der Arbeit wird ein allgemeiner Vergleich zwischen Weblogs und Content Management Systemen erarbeitet. Dabei werden hauptsächlich die Eigenschaften, Einsatzvoraussetzungen, technische Hintergründe sowie praxisnahe Studien und Vor- bzw. Nachteile beider Systeme verglichen. Darauf aufbauend wird im Kapitel 3, das Weblogssystem WordPress mit seinen geschichtlichen Entwicklungen, Charaktermerkmalen sowie aktuellen Funktionen erörtert und Schnittstellen mit einem Web Content Management System aufgezeigt. Besonders der Aufbau und die Funktionsweise der Administrationsoberfläche steht hierbei im Vordergrund. So werden beispielsweise die wichtigsten Bereiche des Backend beleuchtet und untersucht. Des Weiteren wird anhand einzelner Beispiele die Programmlogik und die Strukturierung von Templates mittels PHP und eigenen WordPress-Funktionen wie auch spezifischen Tags, demonstriert.

Das praxisorientierte Kapitel 6 bekräftigt die theoretisch erworbenen Kenntnisse der vorangegangenen Kapitel und verdeutlicht die Umsetzung eines Webprojektes unter den zuvor erarbeiteten Aspekten. Abschließend vervollständigt ein direkter Vergleich zwischen WordPress und dem Web Content Management System TypoLight die Ziele der Arbeit.

1.3 Methodik und wissenschaftliches Vorgehen

Untersuchung hinsichtlich der Nutzbarkeit von WordPress als Web Content Management System beziehen sich zum einen auf aktuelle Studien und zum anderen auf ausgewiesene Fachliteratur und Praxisbeispiele. Unterstützend werden eigene empirische Erkenntnisse herangezogen. Sämtliche Funktionen und Quellcode-Eigenschaften, sowie Analysen über die Implementierung von Skripten werden über die öffentliche Dokumentationsgrundlagen der Betreiberseite¹ bezogen. Für die korrekte und funktionstüchtige Ausführung von Quellcode wird der vom W3C² zur Verfügung gestellte Validator genutzt. Darüber hinaus findet eine Verifizierung aller Funktionen und Erläuterungen in einer eingerichtete Testumgebung³ statt.

1 Siehe: http://codex.wordpress.org/Main_Page

2 Siehe: <http://www.w3.org/>

3 Siehe: <http://entwicklung.rootsofm.de/>

2. Weblogs und Content Management Systeme

In diesem Kapitel wird auf die wesentlichen Unterschiede zwischen Weblogs und Web Content Management Systemen eingegangen. Dabei soll vor allem analysiert werden, was die Systeme differenziert und welche Besonderheiten der jeweiligen Systeme zutreffend sind. Zusätzlich wird darauf geachtet, in welchen Umgebungen Weblogs eingesetzt werden und ab welchem Zeitpunkt die Nutzung eines Web Content Management System sinnvoll ist.

2.1 Weblogs

2.1.1 Allgemein

Der Begriff Weblog oder kurz Blog, ist ein Neologismus und stammt ursprünglich aus dem Englischen. Der Begriff setzt sich aus den Wörtern Web (Netz) und log (Logbuch) zusammen und ist eine Art virtuelles Tagebuch oder Online-Journal, bei denen der Autor seine Inhalte in einem Weblog-System speichert, verwaltet und publiziert⁴.

Die Inhalte, sogenannte POSTS, üblicherweise gespeichert als ARTIKEL oder BEITRÄGE, werden meist in umgekehrter, chronologischer Reihenfolge im Frontend der betriebenen Website dargestellt und dem Leser übermittelt. Bei einem Großteil der im Web zu findenden Blogs besteht für den Leser zudem die Möglichkeit, publizierte Inhalte zu kommentieren und gegebenenfalls zu bewerten. Diese Funktionalität ermöglicht den Austausch von Kenntnissen, sowie eine Weiterentwicklung zukünftiger Themenschwerpunkte.⁵

Das Blog ist ein Medium, welches den Autoren, auch genannt BLOGGER, eine einfach zu handhabende Websoftware bietet, mit deren Inhalte veröffentlicht und diese mit anderen Lesern diskutiert werden können. Grundsätzlich ist das Weblog vergleichbar mit einer persönlichen Netzzeitung oder einem vereinfachtem Webforum⁶.

Neben der Kommentarfunktion von Weblogs sind auch PERMALINKS eine wichtige Eigenschaft von Blogs⁷. Jeder Artikel oder Beitrag bekommt einen langlebigen und eigenständigen Verweis, unter der er gelistet beziehungsweise erreichbar ist. Die referenzierten Beiträge stehen somit immer unter einheitlichen URLs und bleiben bei erneut publizierten Beiträgen bestehen⁸. Dies ist beispielsweise vorteilhaft für den Suchmaschinenoptimierungsprozess, kurz SEO, und bietet zudem die Möglichkeit, dass kurze URLs sich besser einprägen lassen.

4 vgl. Löwer, 2006 [www-dokument]

5 vgl. Alby, 2008a, S. 2

6 vgl. Alby, 2008a, S. 2

7 vgl. Alby, 2008a, S. 2 f.

8 vgl. Sauer, 2007, S.36

PERMALINKS werden deshalb auch oft als ‚sprechende URLs‘ bezeichnet. Auch der Einsatz von PERMALINKS ist bei der Sicherung von BOOKMARKS, Favoritenseiten im Browser, hilfreich und erleichtert unnötiges suchen. Im Zusammenhang mit PERMALINKS spielen auch die so genannten TRACKBACKS eine wesentliche Rolle.⁹

TRACKBACKS vermitteln den Autoren, ob sich ein BLOGGER in einem anderen Blog auf den beschriebenen Beitrag bezieht und verlinkt. Es entstehen somit nicht nur Diskussionen im Blog des Verfassers, sondern auch über die Grenzen des jeweiligen Blogs hinaus. Man spricht in diesem Zusammenhang auch von einer BLOGOSPHERE, einen Zusammenschluss und Verbindung aller Blogs in einem virtuellen, vernetzten Raum.¹⁰

Eine ähnliche Eigenschaft weist auch der sogenannte BLOGROLL auf. Der BLOGROLL ist eine spezielle Art von Verlinkungen innerhalb eines Blogs. Durch die externen Verlinkungen zu anderen Blogs, Webseiten oder Medien kann der Leser sehr schnell die Interessen des BLOGGERS beurteilen und findet somit neue und weiterführende Informationen zum Thema. Einige Blogs beschäftigen sich ausschließlich mit der Verlinkung auf andere Seiten und Beiträge.¹¹

Es gibt im Web eine Vielzahl von Blogs mit unterschiedlichen Eigenschaften und Themen. Beispielsweise wären hier zu nennen die ‚Erzählblogs‘, bei denen der Autor in persönlicher Weise über sein Erlebtes berichtet. Diese ‚Erzählblogs‘ oder auch ‚private Blogs‘ sind im Allgemeinen Web-Tagebücher¹². Im Bereich Web- und Softwareentwicklung gibt es eine große Anzahl an informativen ‚Fachblogs‘. Diese Art der Blogs setzen sich häufig mit speziellen Themenbereichen auseinander, wobei die Autoren über das nötige Hintergrundwissen verfügen, um das geschilderte Thema dem Leser in verständlicher Art und Weise zu erläutern.

Einen weiteren Großteil der Blogkultur nehmen ‚Medien- und Journalistenblogs‘ ein. Eine Vielzahl der Autoren sind renommierte Journalisten oder Redakteure und schreiben, in ihren zum Teil auch persönlichen Blogs, über aktuelle politische und wirtschaftliche Ereignisse. Ein gern genanntes Beispiel für einen erfolgreichen Medienblog ist das BILDblog¹³ von Stefan Niggemeier der bereits im Jahr 2005 den ‚Grimmer Online Award‘ für seinen Blog erhielt. In diesem Blog schreibt der Journalist über das journalistische Handwerk und die dabei unterlaufenden Fehler der Bild-Redaktion bei der Recherche¹⁴. Auch ‚Corporate Blogs‘, ‚Photoblogs‘, ‚Portfolioblogs‘, ‚Tutorialblogs‘ etc. erreichen eine immer größere Popularität.¹⁵

⁹ Vgl. Wöhrer, 2006 [www-dokument]

¹⁰ Vgl. Herring et al., 2005, S. 1f.

¹¹ Vgl. Bültge, 2007, S. 14

¹² Vgl. Bültge, 2007, S. 14 ff.

¹³ Siehe: <http://www.bildblog.de/>

¹⁴ Siehe: Grimme Online Award 2005

¹⁵ Vgl. Alby, 2008b, S. 21 f.

Viele Weblogs bieten des Weiteren ihre Inhalte via NEWSFEEDS an. NEWSFEEDS, kurz FEEDS, sind vergleichbar mit Nachrichtenticker und generieren sämtliche Inhalte des Weblogs in kurzen oder vollständigen Auszügen. Der Vorteil dieser FEEDS ist, dass Nutzer nicht jeden Tag wiederholt auf das Weblog zugreifen müssen, um neue Inhalte zu erhalten. Mit Hilfe eines Programms, den sogenannten FEEDREADER, kann der Leser bestimmte FEEDS und Themen des Blogs abonnieren und wird damit über aktuelle Neuigkeiten informiert. NEWSFEEDS wie beispielsweise RSS oder ATOM basieren auf der Auszeichnungssprache XML und ermöglichen, Inhalte einer Website in eine für Maschinen lesbare Form zu übermitteln¹⁶. Jedoch werden NEWSFEEDS nicht nur bei Weblogs eingesetzt, auch bei Nachrichtenseiten, wie tagesschau.de oder spiegel-online.de kommen FEEDS immer häufiger zum Einsatz¹⁷.

2.1.2 Technik und Software

In der Regel basieren Weblogs auf kleinen Web Content Management Systemen, Webinhaltsverwaltungssysteme, die im nachfolgenden Kapitel näher erläutert werden. Nach KANTEL¹⁸ gibt es im Wesentlichen drei Möglichkeiten Weblogs zu betreiben

1. Die Weblogsoftware läuft auf einem zentralen Server, der gleichzeitig den erforderlichen Webspace bereitstellt, auch HOST genannt. Diese Blog-Dienste wie beispielsweise ‚blogger.com‘ von GOOGLE, ‚myblog.de‘ oder ‚today.net‘ zählen zu den Geläufigsten im BLOGGER-Umfeld. Viele dieser Dienste sind kostenfrei und finanzieren sich vorrangig durch Werbeeinnahmen oder durch kostenpflichtige Upgrades und zusätzliche Serviceleistungen. Ein großer Vorteil solcher HOSTER ist, dass der BLOGGER sofort nach der Registrierung und kurzer Einführung in die Software, beginnen kann, Beiträge zu veröffentlichen. Ein Nachteil kann jedoch sein, dass gerade kostenfreie Anbieter ihre Plattformen schließen und die Nutzer mit ihren Inhalten umziehen müssen¹⁹. Ein solches Szenario ergab sich zum Beispiel am 19. Juli 2009 von 360 GRAD, einen Blog-Service von YAHOO.²⁰
2. Die Blogsoftware wird von einer autorisierten Person oder einem Webmaster auf einen Webserver aufgespielt und anschließend konfiguriert. Alle Anpassungen bezüglich des Bloglayouts, Funktionsweisen, Module und Erweiterungen liegen in der Hand des BLOGGERS beziehungsweise des Administrators. Der bekannteste Vertreter für Blogsoftware in diesem Bereich ist das leistungsstarke WORDPRESS. Aber auch MOVABLE TYPE, TEXTPATTERN

16 Vgl. Simovic, 2007, S. 23

17 Siehe: <http://www.tagesschau.de/xml/rss2> et <http://www.spiegel.de/schlagzeilen/index.rss>

18 Jörg Kantel, *1953, Lehrbeauftragter an der Fachhochschule für Technik und Wirtschaft (FHTW) Berlin im Bereich „Angewandte Informatik“

19 Vgl. Kantel, 2003, S. 3

20 Siehe; <http://help.yahoo.com/l/us/yahoo/geocities/360/>

und `EXPRESSION ENGINE` sind in diesem Zusammenhang erwähnenswert. Die Blogsoftware basiert in den meisten Fällen auf SQL-Datenbanken und der serverseitigen Programmiersprache PHP, welche unter anderem dynamische HTML-Seiten erzeugt.²¹

3. Eine weitere, jedoch nicht so häufig verwendete Möglichkeit, ist der Einsatz der Blogsoftware, die auf dem Rechner des Benutzers installiert ist und statische Seiten erzeugt. Diese werden via FTP auf den Provider hochgeladen und publiziert. Der Vorteil dieser Variante ist vor allem, dass beim Provider keine zusätzliche Software installiert werden muss.²²

2.1.3 Blogstudie

Die Ergebnisse einer Studie, geführt durch Prof. Dr. Ansgar Zerfaß und Janine Bogosyan, der Universität Leipzig veranschaulichen, dass man Blognutzer in nachfolgenden Typologien einordnen kann²³. Im Rahmen der Studie wurden 605 Nutzer befragt, wobei 96 Prozent der Befragten mehrmals täglich im Internet verkehren, 87,3 Prozent schon einen Weblog genutzt haben und sogar jeder Dritte (34,6 Prozent) selbst ein Weblog betreibt und verfasst²⁴.

Es wurde im Einzelnen ermittelt, dass man Blognutzer in fünf verschiedenen Kategorien unterscheiden kann. Dabei wurden 17,7 Prozent der Befragten als „Sozial Networker“ eingeteilt. Diese Gruppe nutzt Weblogs bevorzugt, um in Kontakt mit Freunden und Bekannten zu treten oder um neue Bekanntschaften über ihre Weblogs zu knüpfen. Ebenfalls 17,7 Prozent der Befragten wurden in die Kategorie „Selbstdarsteller“ eingestuft. Die Personen der Kategorie zählen zu den aktiven `BLOGGERN`. Sie werden in der Studie unter anderem damit zitiert, dass sie „etwas zu sagen hätten“ und, dass sie „andere Menschen aufklären“ bzw. „ihr Wissen mit anderen teilen“ wollen. Außerdem ermittelte man, dass 23,7 Prozent der Befragten als „Wissensdurstige“ identifiziert wurden. Diese Gruppe nimmt den größten prozentualen Anteil ein. Die Personen dieser Kategorie nutzen Blogs vorrangig für Hintergrundinformation und trauen klassischen Medien weniger als vergleichsweise Personen anderer Kategorien. Zwei weitere Gruppen sind mit 18,9 Prozent die „Informationssuchenden“ und 22,8 Prozent die „aktiven Konsumenten“. Die Befragten der beiden Kategorien sind vorwiegend auf der Suche nach aktuellen Nachrichten und Produktinformationen. Dabei bestehen laut Ergebnis der Studie die beiden Gruppen vorwiegend aus Bloglesern und seltener aus Autoren.²⁵

²¹ Vgl. Kantel, 2003, S.3 f.

²² Vgl. Kantel, 2003, S.3 f.

²³ Vgl. Zerfaß et al., 2007, S.1ff.

²⁴ Vgl. Zerfaß et al., 2007, S.2 f.

²⁵ Vgl. Zerfaß et al., 2007, S. 8

Der Einfluss von Weblogs auf die öffentliche Meinung ist weitgehend kontrovers diskutiert. Mehr als die Hälfte der Blognutzer (55,4 Prozent) meinen, dass die Inhalte, Einfluss auf die öffentliche Meinung der Leser haben. 53,8 Prozent behaupten sogar, dass Blogs gesellschaftliche Veränderungen herbeiführen können. Demgegenüber widersprechen 41,1 Prozent der Befragten dieser Aussage.²⁶

Die Gründe für die unterschiedlichen Aussagen, bezogen auf die Bildung der öffentlichen Meinung, könnten daran liegen, dass Blogs je nach Urheber eine differenzierte Wertigkeit und Bedeutung auf den Rezipienten auswirken. Für einen Großteil der Befragten sind ‚fachbezogene Blogs‘ von Experten (69,3 Prozent) und Blogs von Medienvertretern, sowie Journalisten (59,5 Prozent) ein wichtiger und eventuell auch unabhängiger Impulsgeber, Themenlieferant und Meinungsbildner. Ebenfalls häufig nutzen 51 Prozent der Befragten ‚private Blogs‘. Nur 12,1 Prozent stellen dabei den Inhalt von ‚privaten Blogs‘ als unglaubwürdig in Frage. Noch deutlicher wird dieses Resultat bei ‚fachspezifischen Blogs‘. Hierbei halten nur 2,1 Prozent die Inhalte für unglaubwürdig und nicht vertrauenswürdig. Häufiger wurde die Glaubwürdigkeit sogenannter ‚Corporate Blogs‘, firmenbezogener Blogs, in Frage gestellt. Ein Viertel der Befragten gab an, den Inhalten firmengeführter Blogs nicht zu trauen.²⁷

2.1.4 Vorteile von Weblogs

Trotz des teilweise auftretenden Missvertrauens gegenüber Weblogs sprechen viele Vorteile für die zunehmende Bedeutung. Eines der größten Vorteile von Weblogs ist die Übersichtlichkeit und die einfache Bedienung (Usability) des Systems. Viele nicht kommerzielle Systeme bieten eine gut strukturierte Benutzeroberfläche und ermöglichen eine einfache Handhabung. Der BLOGGER kann ohne spezielle Programmierkenntnisse Inhalte verfassen. Nach richtiger Konfiguration und regelmäßiger Publizierung neuer Artikel wird durch die PERMALINK-Struktur ein besseres Ranking in Suchmaschinen, als beispielsweise bei vergleichbaren Redaktionssystemen, erreicht. Auch die BLOGOSPHERE und die damit verbundenen ‚Backlinks‘ erzielen eine höhere Positionierung bei Suchmaschinen als herkömmlichen Websites.²⁸ Gegenüber den meisten freien Web Content Management Systemen, deren Seiten- und Kategoriearchitektur festgelegt werden muss, bieten Blogs die Vorteile, dass bereits eine Seiten- und Artikelstruktur vorhanden ist. Dies kann bei vielen Systemen im Nachhinein modifiziert und angepasst werden²⁹. Weitere Vorzüge sind die ständigen Vernetzungen und die damit erreichte Diskussion der BLOGGER untereinander. Die Inhalte der Beiträge werden durch die Leser ständig weiterentwickelt, kommentiert, disku-

²⁶ Vgl. Zerfaß et al., 2007, S. 9

²⁷ Vgl. Zerfaß et al., 2007, S. 10

²⁸ Vgl. Klätke, 2008 [www-dokument]

²⁹ Vgl. Alby, 2008a, S. 3

tiert und verlinkt. Hierdurch wächst nicht nur eine große BLOGGER-Gemeinschaft, es werden auch neue und unabhängige Sichtweisen spezieller Themen beleuchtet. Gerade eine hohe Anzahl an ‚fachspezifischen Blogs‘ unterstützt diese Argumentation³⁰. Jedoch ist die generelle Vertrauenswürdigkeit von Blogs stets zu hinterfragen und eine regelmäßige Verifizierung der Inhalte ratsam. Im Gegensatz zu Zeitungen unterliegen Weblogs und Internet-Enzyklopädien keinem allgemeinem Wahrheitscodex und medienrechtlichen Überwachung. Nur die BLOGGER-Gemeinschaft kann durch Kommentare und Diskussionen fragwürdige Beiträge kenntlich machen.³¹

2.2 Content Management System

2.2.1 Allgemein

Wie der Begriff Weblog, stammt auch der Ausdruck ‚Content‘ aus dem Englischen. Unter dem Begriff werden im Allgemeinen beliebige elektronische Darstellungsformen verstanden. Hauptsächlich wird diese Bezeichnung für den Inhalt numerischer Daten, Texte, Bilddateien, Grafiken, Video- und Audiosequenzen verwendet. Prinzipiell entspricht der Begriff aber jeglicher Art von Dateiinformationen, die auf einem Computer gesichert werden können³². Ein Content Management System verwaltet demnach Medieninhalte, wobei eine strikte Trennung von Inhalt, Layout und Programmierung der Seite besteht.³³

Ein solches System kann beispielsweise eine freie Internet-Enzyklopädie wie WIKIPEDIA oder eine Auktionsplattform wie EBAY sein. In beiden Fällen werden Medieninhalte verwaltet³⁴. Content Management Systeme, kurz CMS, unterstützen in diesem Zusammenhang die ständige Aktualisierung, Archivierung und Aufbereitung von Beiträgen. Jedoch müssen diese Systeme nicht unbedingt etwas mit der Thematik Web- und Webdevelopment zu tun haben. Es gibt eine Vielzahl von Begriffen, die in der Literatur unter der Kategorie Content Management Systemen geführt werden. Vor allem spezielle Arten, wie zum Beispiel ‚Enterprise Ressource Planing System‘, ‚Customer Relaitonship System‘ oder auch ‚Human Ressource Management Systems,‘ werden in diese Gruppe eingebunden. Viele Content Management Systeme werden daher auch als Redaktionssysteme bezeichnet. Wenn diese im Internet Anwendung finden, betrachtet man sie als Web Content Management Systeme (WCMS). Die Verwaltung und Organisation wird bei einem WCMS innerhalb des Webbrowsers beim Client durchgeführt.³⁵

30 Vgl. Alby, 2008a, S.3 f.

31 Vgl. Presserat [www-dokument]

32 Vgl. Lotze et al., 2005, S. 1

33 Vgl. Kemper et al., 2006, S. 129

34 Vgl. Graf, 2008, S. 27

35 Vgl. Graf, 2008, S. 27 f.

Die Kernbereiche solcher Systeme bestehen meist aus mehreren `TEMPLATES`, welche für die Strukturierung der Inhalte, aus einer Datenbank, und die Trennung der Programmierstrukturen verantwortlich sind³⁶. Mit Hilfe dieser Trennung können mehrere Layout-Vorlagen (`TEMPLATES`) im Corporate Design erstellt und dynamische Inhalte aus einer Datenbank ausgegeben werden. Gerade Webdesigner und Grafiker behalten durch `TEMPLATE ENGINES` ihre Freiräume beim Gestalten von Webauftritten innerhalb eines WCMS und kommen somit nicht unnötig mit essentiellen Programmcode- und -strukturen in Verbindung. Designer sind prinzipiell daran gebunden, stilistische Details im Gesamtkonzept zu integrieren und ihr Hauptaugenmerk auf den Frontendbereich der Seite zu konzentrieren. Im Gegensatz dazu bieten `TEMPLATES` auch dem Webentwickler saubere, klar getrennte Arbeitsräume für die Funktion des Systems. Vorzugsweise sollte dabei das Backend und die Systemstrukturierung im Vordergrund stehen.³⁷

Web Content Management Systeme ermöglichen die benutzerfreundliche und angenehme Verwaltung größerer Webauftritte, welche bei der Pflege ursprünglicher Web-Publishing Methoden Probleme bereiten können. Dabei ist vor allem die Benutzer- und Rollenverwaltung eine WCMS vorteilhaft und sorgt so für Effektivität bei der Implementierung von Inhalten. Die Besonderheit der Rollenvergabe ist nicht nur eine Erweiterung der Sicherheit des Systems, sondern ermöglicht, dass eine zusätzliche interne Kommunikation zwischen den einzelnen Akteuren stattfinden kann.

Durch verschiedene Sicherheitsrichtlinien wird in einem WCMS dafür gesorgt, dass nur bestimmte autorisierte Personen, Dokumente erstellen, freigeben, betrachten, archivieren und einsehen können. Außerdem führt eine saubere Trennung der Benutzer, Rollen und Gruppen zu einer Verbesserung und Flexibilität im Umgang mit vertraulichen Datensätzen³⁸. Des Weiteren benötigen Autoren und Redakteure, die Inhalte in ein solches System einpflegen beziehungsweise freigeben, keine Kenntnisse in webüblichen Markup-Sprachen wie zum Beispiel HTML. Der Content wird direkt über eine Eingabemaske eingepflegt und in eine Datenbank gespeichert.³⁹

Ebenfalls ist die Zuverlässigkeit bei Web Content Management Systemen ein charakteristisches Merkmal. Durch die Sicherung systematischer Informationen wird Unordnung und Desorganisation vermieden, wie sie bei der Applizierung gegensätzlicher und alter Datendokumente entstehen kann. Publikationsprozesse bieten die Option, die Erarbeitung und Herausgabe von Inhalten, unter Berücksichtigung der Rollenverteilung, zu automatisieren. Hierbei werden befähigte Personen informiert, welche Tätigkeiten andere Benutzer vollziehen und welche Aufgaben

³⁶ Vgl. Walter, 2008, S. 284

³⁷ Vgl. Nix et al., 2005, S. 20

³⁸ Vgl. Lotze et al., 2005, S. 4

³⁹ Vgl. Nix et al., 2005, S. 20

weiterhin aufgeteilt werden müssen. Auch die Bereitstellung eines Archives macht Informationen zum Content zugänglich und zeigt, in wie weit Datenbestände einer Aktualisierung bedürfen oder gegebenenfalls gelöscht werden können.⁴⁰

2.2.2 Technik und Software

Web Content Management Systeme erfreuen sich bei Unternehmen als auch bei Privatpersonen immer größerer und wachsender Beliebtheit. Immer mehr statische Seiten werden auf Grund ihres Umfangs und der zunehmenden Mitwirkung verschiedener Autoren auf WCMS-Lösungen umgestellt. Dies kann nicht nur ökonomische Vorteile bieten, sondern auch zeitliche und strukturierte Effizienz erzeugen. Entsprechend groß ist das Angebot an WCMS, sowohl in kommerziellen als auch im Open Source Segment. Eine sehr gute Zusammenfassung und Vergleichsmöglichkeit der jeweiligen Systeme bietet die Website www.msmatrix.org, die detailliert in Gegenüberstellungen die Funktionalitäten beleuchtet.⁴¹

Es gibt bei WCMS-Lösungen viele unterschiedlich verwendete Webtechnologien, sowohl im Open Source als auch im kommerziellen Bereich. Darunter zählen unter anderem Systeme, die mit JAVA, ASP.NET, PERL, PYTHON, RUBY ON RAILS, PHP oder vergleichbaren Techniken realisiert werden. Ein Großteil der Open Source WCMS werden unter der Verwendung der Webprogrammiersprache PHP und MySQL-Datenbanken veröffentlicht. Erfahrungsberichte zeigen, dass vor allem Open Source Systeme wie TYPO3, JOOMLA, DRUPAL, CMS MADE SIMPLE, PLONE und TYPOLIGHT im deutschsprachigen Raum in der Entwicklergemeinschaft mit zu den Beliebtesten zählen. Gründe hierfür können sicherlich die GPL-, GNU-, AGPL- Lizenzen sein, als auch der quelloffene Code. Zudem finden Nutzer eine Vielzahl an Entwicklergemeinschaften und optimale Dokumentationsvorlagen auf den jeweiligen Betreiber- und Projektseiten. Ebenfalls wächst die Anzahl der Fachblog- und Buchautoren, die sich der Thematik widmen und somit Hilfestellung bieten.

2.2.3 Anforderung an Web Content Management Systeme

Eine kurze Checkliste soll im Folgenden verdeutlichen, welche Anforderungen an ein WCMS gestellt werden können. Es wird mit Sicherheit kein System geben, gerade im Open Source Bereich, die jede aufgeführte Anforderung vergegenwärtigen. Ebenfalls kann man keine pauschalen Angaben machen, welches das Beste oder das Nutzerfreundlichste ist. Vielmehr sollten Webdesigner- und entwickler die Möglichkeiten der verschiedenen Systeme prüfen und ein, für das Projekt Geeignetes auswählen. Hierfür können dennoch folgende Richtlinien und Anforderungen hilfreich sein.

⁴⁰ Vgl. Lotze et al., 2005, S. 3 ff.

⁴¹ Vgl. Walter, 2008, S. 552

Einfache Bedienbarkeit: Webauftritte sollten von genau den Mitarbeitern gepflegt werden, die über das nötige systemspezifische Fachwissen verfügen und eine kurze Einweisung in das System erhielten. Vor allem Nutzer, die keine speziellen Programmierkenntnisse besitzen, können somit das System bedienen und pflegen.⁴²

Einheitliches Design: Webpräsentationen sind digitale Visitenkarten von Unternehmen, Privatpersonen, sowie Produkt- und Dienstleistungen. Sie sollen beim Besucher Vertrauen und Interesse erzeugen, wobei das Erscheinungsbild stets bewahrt bleiben soll. Es ist erforderlich, dass ein WCMS bei Änderungen und Aktualisierung das vorgegebene Design einhält, auch wenn mehrere Verfasser die Inhalte der Seite pflegen.⁴³

Hohe Integrationsfähigkeit: Ebenfalls wichtig ist die hohe Flexibilität und die Verknüpfungen verschiedenster Dateiressourcen. Offene Schnittstellen sind für das WCMS ausschlaggebend, um mit konträren ERP-, CRM-, SCM-, Shop- und Branchen-Lösungen, ohne größeren Programmier- und Anpassungsaufwand, korrespondieren zu können. So lassen sich beispielsweise Produktdaten aus einem Produktplanungssystem mit medialen Inhalten wie Texten, Bildern, Video- und Audiosequenzen, aus unterschiedlichen Quellen, implementieren und publizieren.⁴⁴

Schlanker Bedarf an IT-Ressourcen: Webauftritte, die mit WCMS-Lösungen realisiert werden, sollten auch ohne Hilfe von IT-Experten ausbaufähig optimiert und zu managen sein. Besonders dann, wenn für die unternehmensweite Administration verschiedene, externe Dienstleister zu Rate gezogen werden müssen.⁴⁵

Mehrsprachigkeit: Vor allem Unternehmen und Dienstleistungen, die international agieren, sollten beim Ausbau eines WCMS auch auf die Mehrsprachigkeit des Systems achten. Wichtig dabei ist, dass der Webauftritt in mehreren Sprachen, sowohl im Backend als im Frontend, effizient pflegbar ist und mit weiteren Sprachen ausgestattet werden kann. Wichtige Sprachmodule sind vordergründig Deutsch, Englisch, Französisch, Italienisch, Russisch und Chinesisch.⁴⁶

Vertrauenswürdige Hersteller: Die Anforderungen an neue Technologien entwickeln sich immer rasanter. Aus diesem Grund sollte ein WCMS mit neuen technologischen Ansprüchen und systembezogen Updates kompatibel sein, um somit Aktualität und Einhaltung definierter Webstandards zu gewährleisten. Daher ist es ratsam, bei der Auswahl von kommerziellen Systemen

⁴² Vgl. Onasch, 2006 [www-dokument]

⁴³ Vgl. Onasch, 2006 [www-dokument]

⁴⁴ Vgl. Onasch, 2006 [www-dokument]

⁴⁵ Vgl. Onasch, 2006 [www-dokument]

⁴⁶ Vgl. Onasch, 2006 [www-dokument]

einen finanzkräftigen Anbieter und bei Open Source Lösungen, etablierte und vom Nutzer favorisierte WCMS, die auch langfristig Updates und Support gewährleisten, zu wählen.⁴⁷

Überschaubare Kosten: Viele Internet-Dienstleister versuchen mit besonders günstigen Einstiegspreisen Neukunden zu gewinnen. Im Projektverlauf kann es jedoch zu unerwartet hohen Folgekosten durch Anpassung einzelner Module kommen. Genauere Kostenvoranschläge mit Minimum- beziehungsweise Maximumszenarien helfen, unerwartete Kosten frühzeitig aufzudecken und gegebenenfalls zu vermeiden. Auch die Erstellung eines Pflichtenheftes ist in dem Zusammenhang von höchster Priorität.⁴⁸

Barrieren vermeiden: Um Web- und Industriestandards einzuhalten, sollten alle gewählten WCMS auf Barrierefreiheit großen Wert legen. Es ist wichtig, dass WCMS die Voraussetzungen besitzen, fehlerhafte Hyperlinks darzustellen, Bilder korrekt zu betiteln und Texte fehlerfrei abbilden zu können. Anhand der Dokumentation und Auswertung können Fehler vermieden werden, bevor sie für den Endnutzer sichtbar sind.⁴⁹

Einfache Erweiterung: Ein gutes und flexibles WCMS bietet die Möglichkeit der modularen Erweiterung an die vom Kunden gewünschten Bedürfnisse. Beispielsweise sollte ein WCMS im Zuge des Unternehmenswachstums mitwachsen können, um investitionssicher zu sein und die Anforderung von Management und Mitarbeitern gleichmäßig zu gewährleisten. Vor allem neue Applikationen und PLUGINS sollten sich gut integrieren lassen.⁵⁰

Sicherer Schutz: Web Content Management Systeme sind in der Regel unternehmenskritische Anwendungen. Daher ist der Schutz vor Fremdzugriffen unerlässlich. Zum einen kann man sich durch ein gut strukturiertes und vor allem nach aktuellen Standards programmiertes System, und zum anderen durch eine optimale organisierte Benutzer- und Zugriffsverwaltung innerhalb des Systems, vor Angreifern von außen schützen. Wie stark einzelne WCMS diese Zugriffsberechtigung abstufen, kann ein Evaluierungskriterium für die Anforderung des Systems sein.⁵¹

47 Vgl. Onasch, 2006 [www-dokument]

48 Vgl. Onasch, 2006 [www-dokument]

49 Vgl. Onasch, 2006 [www-dokument]

50 Vgl. Onasch, 2006 [www-dokument]

51 Vgl. Nix et al., 2005, S. 139 f.

2.2.4 Zusätzliche Systemfeatures von Web Content Management Systemen

- WYSIWYG-Editor und zusätzliche Erweiterung mit HTML-Editor
- MS Word / Texteditor Integrationsfähigkeit
- Formatierungsmöglichkeiten für erstellte Texte
- Tabelleneditor zur Erstellung valider Tabellen
- PDF-Dokumentenerstellung
- Erstellung von Druckdateien der angebotenen Inhalte
- Import- und Exportfunktionalität weiterer Medieninhalte
- Funktion zur Bildbearbeitung wie beispielsweise das Skalieren von Bildern
- Zeitgesteuertes Veröffentlichen von Artikeln und Beiträgen
- Formulargenerator
- Kommentarfunktionen bei Artikeln
- Spamschutz
- Archivfunktion
- Sitemapgenerator
- Suchfunktion innerhalb des Systems
- Skalierbarkeit
- Mehrsprachigkeit
- Zugriff- und Nutzerverwaltung
- Rechteverwaltung
- Hyperlinkkontrolle
- XML-Schnittstellen
- FEEDS zum Beispiel RSS oder ATOM
- SEO-Schnittstellen
- ToDo-Listen und Aufgabenverteilungen
- Backup-Funktion sowie Update-Fähigkeiten
- Visuelles Feedback auf Javascript-Basis, wenn im Hintergrund Aktionen laufen
- Hilfefunktion und Support
- umfangreiche Dokumentation
- Multiuser mit der Verwaltung mehrerer Websites

2.2.5 WCMS-Studie

Die Ergebnisse, der im Rahmen des FRAUNHOFER INSTITUTS und TIKOM durchgeführten WCM Studie, verdeutlichen, dass die praktische Bedeutung von Webauftritten für ein Unternehmen weniger von der Größe, als von der Branche abhängig ist. Während Dienstleister wie Handwerksunternehmen, Industrie- und Verkehrsunternehmen die eigene Internetpräsenz tendenziell nachrangig betrachten, sehen Tourismus, IT- und Telekommunikationsunternehmen ihre Website als Stützpunkt ihrer Marketingstrategie. Des Weiteren wurde ermittelt, dass acht Prozent der Um-

frageteilnehmer, vorwiegend aus der IT- und Kommunikationsbranche, sowie Industrie und dem öffentlichen Dienst, Webseiten mit über 1000 Seiten betreiben. Im Gegensatz dazu beschränken sich 42 Prozent der Befragten, darunter vor allem klein- und mittelständige Unternehmen, auf eine Webpräsenz von weniger als 25 Seiten. Ein Viertel der teilnehmenden Unternehmen erwarten ein Inhaltswachstum von ungefähr 50 Prozent, wobei die IT- und Telekommunikationsbranchen, Industrie und öffentlicher Dienst, den stärksten Zuwachs vorhersehen.⁵²

Ein weiterer Beleg für die Pflegeintensität stellt in diesem Zusammenhang der Änderungsrhythmus und der jeweilige Zeitabstand zwischen den einzelnen Änderungen dar. Immerhin führen 40 Prozent der befragten Unternehmen gewöhnlich mehr als eine Änderung pro Woche an ihrer Website durch. Dabei werden auch hier die Webseiten von IT- und Telekommunikationsunternehmen, sowie öffentlichen Organisationen wesentlich häufiger pro Woche aktualisiert.⁵³

Unternehmen haben aus diesem Grund erkannt, dass das Angebot an hochwertigen und aktuellen Inhalten für die Interaktion mit den Kunden ausschlaggebend ist. Hierdurch steigt die Pflege und der Aufwand bei der Aktualisierung der Internetpräsenz, weshalb WCMS vermehrt zum Einsatz kommen.⁵⁴

Die befragten 780 Unternehmen gaben mit 46 Prozent in der Studie an, ein webbrowsersbasierendes WCMS zur Pflege ihrer Website einzusetzen. Auffällig ist dabei, dass vor allem Tourismusunternehmen, wie beispielsweise Online-Reiseunternehmen, Tourismusverbände oder Hotels, die durch ständige Aktualität der Inhalte enge Kundenbindung erreichen, sich an der Spitze befinden. Auf dem zweiten Platz ordnen sich überraschend die IT- und Telekommunikationsbranchen ein. Auf die Frage, welche Inhaltsbereiche dabei vor allem gepflegt werden, wurden aktuelle Neuigkeiten und Informationen zu den angebotenen Produkten genannt. Jedoch spielen Seiten wie Service und Support, als auch Unternehmensinformationen und Stellenangebote eine geringfügigere Rolle bei der Pflege über ein WCMS. Erfahrungsgemäß bearbeiten und aktualisieren größere Unternehmen und Dienstleister solche Seiten auch über die Verwaltungssysteme. Die Hauptargumente für den Einsatz von WCMS, welche bereits von Unternehmen eingesetzt werden und von jenen, die in Zukunft auf solche Lösungen zurückgreifen werden, sind vorrangig die erreichbare „Aktualität und Qualität“, sowie die „Pflege der Website ohne technische Kenntnisse“. Als weitere Merkmale wurden „Unabhängigkeit von Mitarbeitern und externen Dienstleistern“ sowie „Kostenoptimierung“ genannt.⁵⁵

52 Vgl. Weber, 2002 [www-dokument]

53 Vgl. Weber, 2002 [www-dokument]

54 Vgl. Weber, 2002 [www-dokument]

55 Vgl. Weber, 2002 [www-dokument]

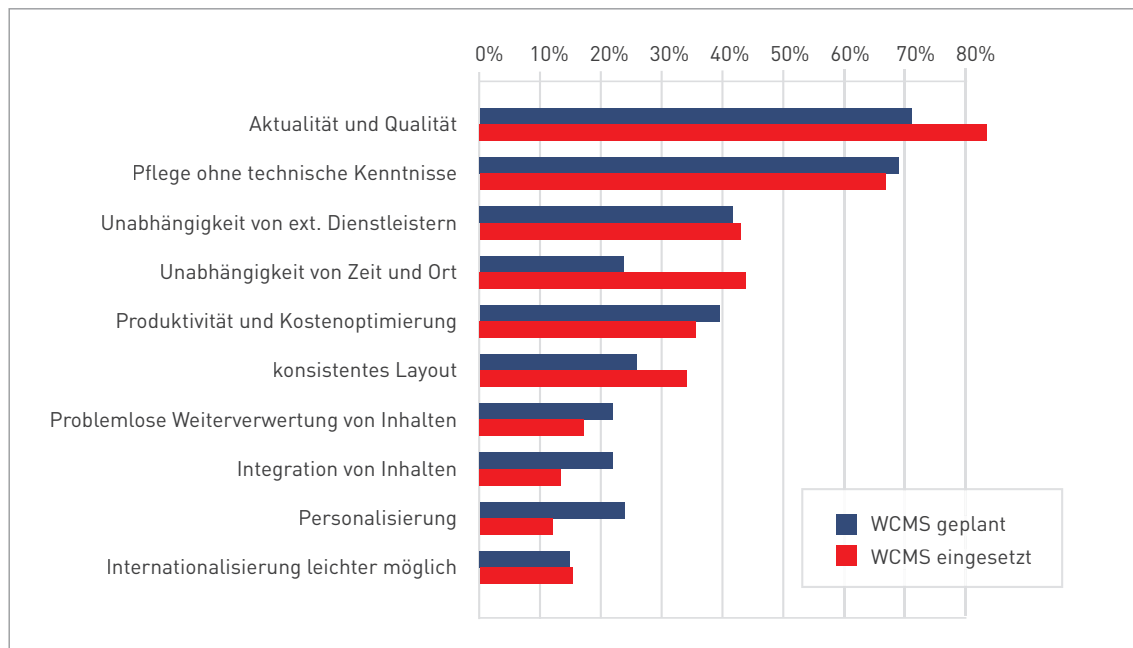


Abbildung 2.1: Ergebnisse der WCMS-Studie, eigene Darstellung⁵⁶

Der Großteil an Unternehmen, die in dieser Studie befragt wurden und noch kein WCMS einsetzen, hält ein solches System, für die eigene Website, nicht notwendig und meint, dass „die aktuelle Situation vollkommen zufriedenstellend ist“. Ein Drittel der Umfrageteilnehmer äußerten sich besorgt über die zeitliche Implementierung des Systems und für ein Viertel sprachen die Investitionskosten gegen die Realisierung.⁵⁷

In den letzten Jahren standen immer wieder zu hohe Investitionskosten für klein- und mittelständigen Unternehmen für die Anschaffung eines WCMS im Weg. Die Mehrzahl der Kleinunternehmer gab in der Erhebung an, bis zu 730 Euro pro Jahr für die Pflege ihre Internetpräsenz auszugeben. Im Vergleich dazu investieren mittelständige Unternehmen pro Jahr 3600 Euro, Großunternehmen bis zu 72.700 Euro. Im Branchenvergleich führt bei den Ausgaben die Geld-, Kredit- und Versicherungsbranche vor IT- und Telekommunikationsunternehmen und öffentlichen Einrichtungen. Aufgrund des enormen Angebots an WCMS-Anbietern, können vor allem klein- und mittelständige Unternehmen von einem optimalen Preis-/Leistungsverhältnis profitieren, da sich viele Anbieter auf diese Unternehmensbranche konzentrieren.⁵⁸

⁵⁶ In Anlehnung an Weber, 2002 [www-dokument]

⁵⁷ Vgl. Weber, 2002 [www-dokument]

⁵⁸ Vgl. Weber, 2002 [www-dokument]

2.3 Zusammenfassung Weblog und Content Management System

Während das Weblog hauptsächlich der chronologischen Abbildung und Darstellungen von publizierten Artikeln in einen festgesetzten Layout dient, können in einen CMS verschiedene Inhalte in unterschiedlichster Form miteinander verknüpft und präsentiert werden. Dies kann vor allem zu komplexeren Navigationsstrukturen führen, welche die Ausgabe von Inhalten in unterschiedliche Bereiche kategorisiert, wobei die chronologische Reihenfolge eher eine untergeordnete oder gar keine Rolle spielt. Die Verwaltung von Inhalten, Nutzern und Layout erfolgt meist über eine Administrationsoberfläche, kurz Backend.⁵⁹

Der Vergleich zwischen Weblog und WCMS ist hier stark vereinfacht. Viele Weblog-Applikationen beherrschen mittlerweile etliche Funktionen, beispielsweise durch Erweiterungen oder PLUGINS, die zu den Standards von WCMS gehören. Auch die Blogsoftware WORDPRESS, die seit den letzten Versionen enorm gewachsen ist und gerade Bereicherung aus dem WCMS-Segment erfahren hat, ist unter diesen Gesichtspunkten mit einem Web Content Management Systemen vergleichbar. So unterstützt WORDPRESS zum Beispiel die Darstellung statischer Seiten unabhängig von der Erstellung und Chronologie, sowie der Verschlagwortung durch TAGS und Kategorisierung.⁶⁰

Ein bereits beschriebenes WCMS-Merkmal ist die Nutzerverwaltung. So erlauben zum Beispiel differenzierte Zugriffsrechte, die Einrichtung und Administrierung komplexer Workflows. Auch in diesem Bereich findet man bei WORDPRESS fortschrittliche Weiterentwicklungen. Seit einigen Versionen ist unter diesem Stichwort ein Rechtesystem mit Rollen und Benutzern integriert. Seit der Version 2.6. sind sogenannte Berechtigungsobjekte und ein Revisionssystem hinzugekommen, welche die Historie eines publizierten Artikels abbilden.⁶¹

Im Allgemeinen ließe sich ein Weblog-System als Spezialfall eines vereinfachten und reduzierten WCMS vergleichen, wobei gerade Funktionen wie FEEDS, PINGS, TRACKBACKS oder KOMMENTARE die Anwendungen in einem herkömmlichen WCMS-Umfeld interessant machen. Viele große Open-Source WCMS haben mittlerweile einige dieser Funktionen in den Kern des Systems integriert und bieten neben dem herkömmlichen System, Blog-Erweiterungen an.⁶²

⁵⁹ Vgl. Bültge, 2007, S. 22f.

⁶⁰ Vgl. Bültge, 2007, S. 22f.

⁶¹ Vgl. Huesmann, 2008, [www-dokument]

⁶² Vgl. hierzu TYPOLIGHT, JOOMLA, DRUPAL, TYPO3

3. Einführung WordPress

3.1 Historische Entwicklung des Begriffs Weblog

Bevor der Begriff Weblog existierte, verfügten einige Webseiten über blogähnliche Aufbaustrukturen, die den heutigen Blogs entsprechen. Es handelte sich dabei um Webseiten, in denen die Autoren ihren Lebensalltag oder Ereignisse schilderten und eine chronologische Darstellung der Inhalte wählten. Zu den ersten **BLOGGER** zählte beispielsweise **DAVE WINER**, der eines der ersten **BLOGGING**-Systeme programmierte und für seine Website **SCRIPTING NEWS** nutzte. Er war außerdem an der Entwicklung von **RSS** und **METAWEBLOG API** beteiligt.⁶³

Ähnlich wie **DAVE WINER** hatten frühere **BLOGGER** eigene Programme zur Veröffentlichung von Artikeln erstellt oder nutzen klassische **HTML**-Werkzeuge. Gleichzeitig entstanden die ersten Services im Web, die das Anlegen und Verwalten eigener Inhalte in einem Blog ermöglichten. Die darauf folgende User-Generation nutzte bereits Ende der 90er Jahre, zum Publizieren von Inhalten, Anbieter wie **OPEN DAIRY**, **LIVEJOURNAL** oder **BLOGGER.COM**. Diese Dienste wurden jedoch nur von einer internetbegeisterten Minderheit genutzt.⁶⁴

Ende 1997 entstand der Begriff Weblog. Dieser Begriff wurde vom Webautor, **JORN BARGER** geprägt. Durch die Modifizierung des Begriffs in „We Blog“ durch **PETER MERHOLZ** im Jahr 1999, entstand hieraus die Bezeichnung „Blog“⁶⁵. Im Jahr 2001 wurde vor allem in den Vereinigten Staaten von Amerika eine breite Öffentlichkeit auf Weblogs aufmerksam.⁶⁶

Es entstanden Blogs, die sich vor allem den Themen Technik und Politik widmeten, wie beispielsweise **POLITICAL WIRE** oder **INSTANPUNDIT**⁶⁷. Zu dieser Zeit ermöglichten viele Bloghoster den Besuchern, Feedbacks in Form von Kommentaren zu hinterlassen. Gerade im Bereich Politik, war dies ein willkommener Fortschritt⁶⁸. Viele Blogs wurden unter diesen Gesichtspunkten immer häufiger als alternative Nachrichtenquelle angesehen. **BLOGGER** transportierten so beispielsweise politische Unstimmigkeiten und berichteten aus Ländern, in denen eine verstärkte Nachrichtenzensur bestand.⁶⁹

⁶³ Vgl. Alby, 2008a, S. 4f.

⁶⁴ Vgl. Alby, 2008a, S. 5f.

⁶⁵ Vgl. McCullagh et al., 2007 [www-dokument]

⁶⁶ Vgl. Alby, 2008a, S. 5f.

⁶⁷ Siehe <http://www.instpundit.com> et <http://www.politicalwire.com/>

⁶⁸ Vgl. Alby, 2008a, 5f.

⁶⁹ Vgl. Alby, 2008a, S. 6

Auch in Deutschland stieg das Interesse an Weblogs stetig an. Traditionelle Medien wie z.B. DIE ZEIT, DER SPIEGEL oder die SÜDDEUTSCHE ZEITUNG integrierten Weblogs in ihr multimediales Angebot⁷⁰.

Im Jahr 2005 wurden in Deutschland Weblogs auch als Kommunikations- und Informationsmittel zur Landtagswahl in Nordrhein-Westfalen eingesetzt. Damit erhielten sie in größerem Umfang politische Bedeutung und fungierten für Parteien, Kandidaten und Bürgern als Diskussion- und Kommunikationsplattform.⁷¹

In der darauf folgenden Bundestagswahl stieg die Zahl der Wahlkampfblogs und politisch motivierten Weblogs weiter an. Es etablierten sich politische Informationsblogs, wie beispielsweise WAHLBLOG05.DE, oder LAUTGEBEN.DE. Einer der ersten Politiker mit eigenem Weblog war der SPD-Politiker und Hamburger-Bürgermeisterkandidat, THOMAS MIROW, im Jahr 2005.⁷²

In den folgenden Jahren 2007 bis 2009 war in der BLOGOSPHERE ein weiterer Trend sichtbar. Microblogging-Dienste, wie TWITTER oder Instant Messaging Dienste, wie SKYPE etablierten sich zunehmend und waren aus der Webentwicklung nicht mehr wegzudenken. Gerade TWITTER, zu deutsch „zwitschern“ vollzog nach Ermittlung einer amerikanischen Studie eine rasante und unglaubliche Entwicklung. Dabei zählte die Studie im Februar 2008, 475000 Nutzer bei TWITTER. Bereits ein Jahr später betrug die Anzahl der registrierten Nutzer 7038000⁷³. Der Microblogging-Dienst ermöglicht es innerhalb kürzester Zeit, Nachrichten mit einer Maximallänge von 140 Zeichen an andere Nutzer (FOLLOWER) zu schicken, um ihnen mitzuteilen, welche Tätigkeiten man gerade ausführt.⁷⁴

Der Autor und BLOGGER SASCHA LOBBO meint auf seinem privaten Blog zu TWITTER: „[...] durch die Beschleunigung und erhöhte Vernetzung per Microblogging gewinnen Blogs deutlich an publizistischem Einfluss. TWITTER fungiert hier als Social Newsticker, dessen Durchschlagkraft von der Relevanz für die Blogszene abhängt, die nicht zufällig weitgehend der TWITTERSZENE entspricht“.⁷⁵

Ähnlicher Auffassung sind in Deutschland auch einige Politiker, die den Microblogging Service für ihre Zwecke entdeckt haben. Zur Bundestagswahl 2009 nahm neben den herkömmlichen Blogartikeln auch TWITTER-Beiträge, kurz TWEETS, von Politikern zu. Nach Untersuchungen der

70 Siehe <http://www.zeit.de/blogs/index> et <http://www.tagesspiegel.de/weblogs> et <http://www.blog.sueddeutsche.de>

71 Vgl. Abold, 2005 [www-dokument]

72 Vgl. Sonnabend, 2005 [www-dokument]

73 Vgl. McGiboney, 2009 [www-dokument]

74 Vgl. Fitton, 2009, S. 1f.

75 Vgl. Lobbo, 2009 [www-dokument]

Analysefirma NIELSEN MEDIA gab es im ersten Quartal 2009 von insgesamt 612 Bundestagsabgeordneten, 68 TWITTER Accounts⁷⁶.

Vor der Bundestagswahl 2009 stieg die Zahl der TWITTER-Accounts bei Politikern und deren Verbänden enorm an⁷⁷. Fraglich ist in diesem Zusammenhang, wie wahrheitsgetreu die jeweiligen Accounts sind und ob eine generelle Strategie hinter der Informationsverteilung via TWITTER steckt. Laut einer Studie der GESELLSCHAFT FÜR STRATEGISCHE KOMMUNIKATION MBH PR-COM aus München sind „Parteien von TWITTER völlig überfordert“ und „[...] eine strategische Nutzung der Plattform für den Wahlkampf ist nicht erkennbar“.⁷⁸

3.2 Historische Entwicklung der Blogsoftware WordPress

WordPress ist ein Weblog-Publishing-System, welches auf die ständige Aktualisierung von Webinhalten in chronologischer Form zugeschnitten ist. Das System bietet schnelle und unkomplizierte Bearbeitungsmöglichkeiten von multimedialen Inhalten, wie das Erstellen und Veröffentlichen von Textbeiträgen. Die Entwickler legten beim Aufbau des Systems eine strikte Trennung von Layout und Programmierung fest, auf die in den folgenden Kapiteln näher eingegangen wird.⁷⁹

Die Applikation WordPress entwickelte sich aus dem Blogsystem B2/CAFELOG von MICHEL VALDRIGHI, welche im Jahr 2001 begann. B2/CAFELOG basierte auf der Programmiersprache PHP und einer MySQL-Datenbank. Die Hauptentwickler von WordPress, MATT MULLENWEG und MIKE LITTLE entwickelten zunächst nach Einstellung des B2-Programmcodes, WordPress auf den B2-Grundlagen-code. Aus diesem Grund wird WordPress auch als offizieller Nachfolger von B2 betrachtet.⁸⁰

Durch Änderung des Lizenzmodells der Blogsoftware MOVEABLE TYPE im Jahr 2004 erlangte WordPress erste Popularität bei den BLOGGERN. Die MOVABLE TYPE Lizenzen waren ab diesem Zeitpunkt komplizierter und die Nutzung nicht mehr kostenfrei⁸¹. Zudem bot WordPress ab Version 1.0 die Möglichkeit, bereits bestehende Inhalte in die Applikation zu implementieren. Kritiker behaupten dagegen, dass die Hauptaufgaben der ersten WordPress Versionen darin lagen,

⁷⁶ Vgl. Trost, 2009 [www-dokument]

⁷⁷ Vgl. Simon, 2008 [www-dokument]

⁷⁸ Vgl. Koerber, 2009 [www-dokument]

⁷⁹ Vgl. Bültge, 2009, S. 23

⁸⁰ Siehe <http://b2evolution.net>

⁸¹ Vgl. Burnette, 2007 [www-dokument]

MOVABLE TYPE Funktionalitäten zu kopieren.⁸² Viele verärgerte BLOGGER suchten aufgrund des geänderten Lizenzmodells nach Ausweichmöglichkeiten und fanden vor allem in WordPress eine Alternative.

Die nachfolgende Abbildung zeigt die weltweite Entwicklung von MOVABLE TYPE und WordPress in den Jahren von 2004 bis 2009.⁸³

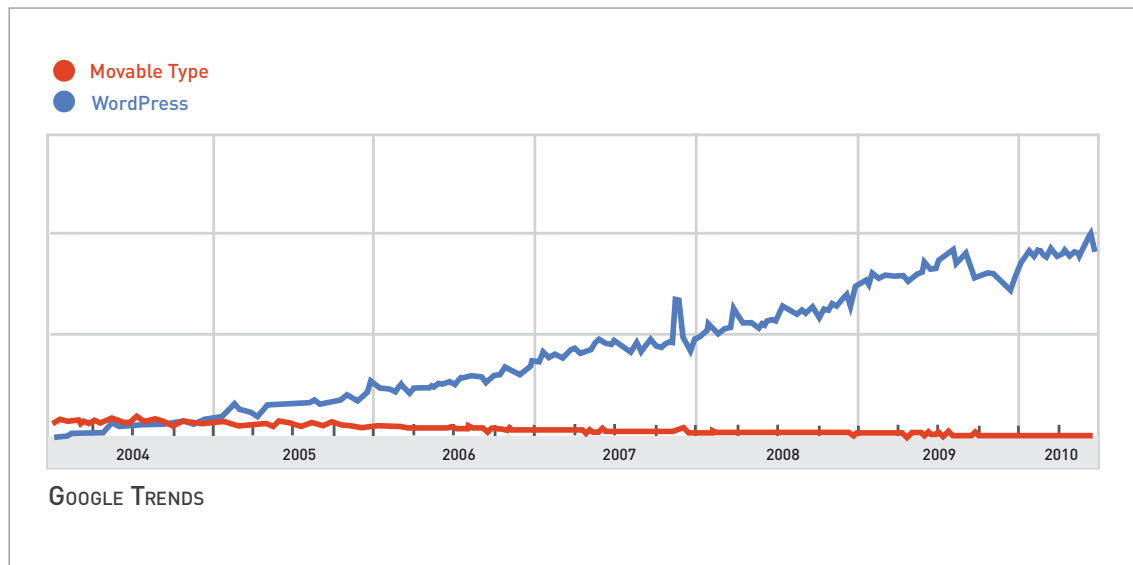


Abbildung 3.1: Nutzerentwicklung von MOVABLE TYPE und WordPress

Alle veröffentlichten WordPress Versionen, beginnend ab Version 1.0, tragen die Namen von bekannten Jazzmusikern. Die aktuelle Version 2.8.4, September 2009, trägt den Name BAKE benannt nach dem Trompeter CHET BAKER. Der Grund für die Titelf Vergabe ist, dass die Entwickler leidenschaftliche Jazzfans sind und den Musikern eine Ehre für ihre Dienste erweisen möchten.⁸⁴

⁸² Vgl. Alby, 2008a, S. 7f.

⁸³ Siehe: <http://www.google.com/trends?q=WORDPRESS%2C+movable+type&ctab=0&geo=all&geor=all&date=all&sort=0>

⁸⁴ Vgl. Mullenweg, 2010 [www-dokument] - Roadmap

3.2.1 WordPress Versionen im Überblick

WordPress 1.0: Die erste Version von WordPress 1.0 erschien im Januar 2004 und enthielt die meisten Funktionen, die auch bei Movable Type implementiert waren.⁸⁵

WordPress 1.01 (Miles Davis): Diese Version erschien kurz nach der ersten Veröffentlichung. Sie trägt als erste den Namen eines Jazzmusikers und beseitigte Fehler der Vorgängerversion. WordPress löste sich mit dieser Version von B2/CAFÉLOG ab und nutzte erstmalig die Installationsdatei ‚wp-config.php‘. Aufgrund der Beseitigung von Programmfehlern, gilt die Version Miles Davis als erste stabile Erscheinung von WordPress.⁸⁶

WordPress 1.0.2 (Art Blakey, kurz Blakey): Ebenfalls wie die Vorgängerversion Miles wurden in der Veröffentlichung Blakey weitere Fehlerbereinigung vorgenommen.⁸⁷

WordPress 1.2 (Charles Mingus): Die im Mai 2004 veröffentlichte Version 1.2 enthielt vor allem die Einbindung von Erweiterskripten und Funktionen (Plugins), Unterkategorien und die Unterstützung von Unicode sowie die Möglichkeit einer Lokalisierung.⁸⁸

WordPress 1.5 (Billy Strayhorn): In der Version 1.5, die am 17. Februar 2005 veröffentlicht wurde, implementierten die Entwickler ein neues und verbessertes Template-System. Dieses ermöglichte den Nutzern die Erstellung sogenannter Themes (WordPress Layoutvorlagen). Des Weiteren wurde zum ersten Mal, neben der Erstellung von Beiträgen (posts), auch die Verwendung einer Seitenerstellung (Seiten) eingeführt, welches WordPress als seitenbasierenden WCMS ermöglicht.⁸⁹

WordPress 2.0 (Duke Ellington): Mit der Veröffentlichung von Duke am 31. Dezember 2005, wurde das Backend von WordPress komplett erneuert. So wurde beispielsweise das Backendlayout modernisiert und ein neuer WYSIWYG-Editor, der auch unter Mac OS X lief, eingebaut. Außerdem erzielte man mit Hilfe von Javascript, ein verbessertes Handling von Administrationswerkzeugen. Auch der zentrale WordPress-Code wurde erneuert, um beim Verfassen von Beiträgen oder Kommentaren, direkten Datenbankzugriff zu erhalten. Mittels Vorschau-Funktion konnten Beiträge und Seiten während des Erstellvorganges im Frontend betrachtet werden.⁹⁰

⁸⁵ Vgl. Mullenweg, 2004 [www-dokument] - WordPress 1.0

⁸⁶ Vgl. Mullenweg, 2004 [www-dokument] - 1.0.1 Release Candidate

⁸⁷ Vgl. Mullenweg, 2004 [www-dokument] - Changelog/1.0.2

⁸⁸ Vgl. Mullenweg, 2004 [www-dokument] - 1.2 Release Candidate

⁸⁹ Vgl. Mullenweg, 2005 [www-dokument] - Announcing WordPress 1.5

⁹⁰ Vgl. Mullenweg, 2005 [www-dokument] - WordPress 2.0

WordPress 2.1 (Ella Fitzgerald): Die Version 2.1 ließ über ein Jahr nach dem Erscheinen von Duke auf sich warten. Es gab unter der Version 2.1 eine verbesserte und neue AJAX-Funktionalität, welche sich positiv auf die Darstellung und Performance im Backend auswirkte. Auch der Programmcode der Datenbank wurde überarbeitet, so dass es hier zu Leistungssteigerungen kam. Zudem wurden viele Sicherheitslücken geschlossen und beim Verfassen von Artikeln, integrierte man eine Zwischenspeicherung.⁹¹

WordPress 2.2. (Stan Getz): Die im Mai 2007 veröffentlichte Version unterstützte vor allem den Widget-Support für Templates und Plugins. Mit dieser Widget-Funktion können in der Sidebar per ‚drag and drop‘ Inhaltsoptionen wie z.B. die letzten Beiträge, die neuesten Kommentare eingepflegt werden, ohne das gesamte Template oder Theme manuell zu bearbeiten. Hinzu kam die vollständige Unterstützung des auf XML-basierende Atom Syndication-Format (Atom 1.0).⁹²

WordPress 2.3 (Dexter Gordon): In der Version Dexter wurde das langerwartete Tagging von Beiträgen eingeführt, welches in den Vorgängerversionen nur durch zusätzliche Plugins möglich war. In der Administrationsoberfläche informiert WordPress darüberhinaus über die neueste Version. Diese Funktion trifft sowohl für Neuerscheinungen von Systemversionen, als auch bei Plugins-Erneuerungen zu. Außerdem wurden wiederholt Sicherheitslücken geschlossen und ein neues ‚Taxonomie-System‘ für Kategorien eingeführt.⁹³

WordPress 2.5 (Michel Brecker): Aufgrund eines hohen Zeitverzugs bei der Entwicklung des WordPress-Cores wurde die Version 2.4 ausgelassen und man veröffentlichte stattdessen am 29. März 2008 die Version 2.5 unter dem Namen Michel Brecker. Diese bot eine neue, überarbeitete Administrationsoberfläche und überzeugte durch gesteigerte Performance. Jedoch stieß die neue Oberfläche auf heftige Kritik bei vielen Nutzern. So waren beispielsweise Arbeitsprozesse beim Erstellen von Beiträgen zeitaufwendiger und teilweise untransparenter als in Vorgängerversionen. Viele notwendige Funktionen waren auf unterschiedliche Stellen im Backend verteilt, so dass eine gewisse Desorganisation bestand. Positiv festzustellen war jedoch, das neu strukturierte Dashboard, womit sämtliche Bereiche des Blogs komprimiert zusammengefasst waren.⁹⁴

91 Vgl. Mullenweg, 2007 [www-dokument] - WordPress 2.1 Ella

92 Vgl. Mullenweg, 2007 [www-dokument] - WordPress 2.2

93 Vgl. Mullenweg, 2007 [www-dokument] - WordPress 2.3

94 Vgl. Mullenweg, 2008 [www-dokument] - WordPress 2.5

WordPress 2.6 (MCCoy TYNER): Da die WordPress-Entwickler für schnelle Veröffentlichungs- und Entwicklungsprozesse bekannt sind, wurde nur drei Monate nach der Version 2.5, TYNER der Öffentlichkeit vorgestellt. Hauptsächlich Änderungen vollzogen sich im Bereich der Sicherheitsarchitektur. Zudem gab es kleine Veränderungen im Backend des Systems. So bot der Administrationsbereich, beim Schreiben von Beiträgen, das Zählen von Wörtern an. Außerdem beinhaltete die Version den TURBO-MODUS, welcher den Benutzern erlaubte, WordPress mit Hilfe von GOOGLE GEARS auch offline zu betreiben. Damit sollten die Leistung und Geschwindigkeit verbessert werden. Ebenfalls bestand seit dieser Version die Möglichkeit, PLUGINS automatisch über das Backend zu installieren.⁹⁵

WordPress 2.7 (JOHN COLTRANE): Die Version 2.7 wurde dem Jazz-Saxophonisten JOHN COLTRANE gewidmet und ist im Vergleich zu allen anderen Veröffentlichungen, diejenige, mit den gravierendsten Änderungen. Vor allem das Backend sticht hier hervor. Dieses wurde vom Design, Benutzerfreundlichkeit und Menüführung komplett überarbeitet. Kleine Icons verdeutlichen anstelle umständlicher Begriffe ein neues, übersichtliches Navigationsmenü.⁹⁶

Weitere Neuerungen vollzogen sich im DASHBOARD. Hier können mit ‚drag and drop‘, Module angepasst und geordnet werden. Das DASHBOARD wurde des Weiteren um nützliche Module erweitert. Vor allem der Mini-Text Editor QUICKPRESS ist an dieser Stelle erwähnenswert. Hier können aus dem DASHBOARD heraus, ARTIKEL erstellt und editiert werden. Außerdem ermöglicht die Version 2.7 die Moderation von Kommentaren aus dem DASHBOARD.⁹⁷

Auch bei der Erstellung von ARTIKELN wurde in dieser Version zukunftsweisend gearbeitet. So können beispielsweise ARTIKEL gleichzeitig in einer MEHRFACHBEARBEITUNG erstellt und editiert werden. Die Einbindung von TAGS wurde ebenfalls erneuert und weist darauf hin, dass bestimmte TAGS bereits vergeben sind. Im Bereich der Artikelerstellung gibt es die Möglichkeit, Beiträge zu einem späteren Zeitpunkt zu veröffentlichen.⁹⁸

Auch die MEDIATHEK wurde teilweise überarbeitet. Medieninhalte können ohne das Erstellen von ARTIKELN oder SEITEN, in die MEDIATHEK eingebaut und zu einem späteren Zeitpunkt verwendet werden. Der Upload funktioniert herkömmlich über den Webbrowser oder via Flashuploader. Insgesamt gibt es durch die Modularisierung von ARTIKEL, SEITEN, KOMMENTARE eine verbesserte Übersicht des Systems und deren Inhalte.⁹⁹

⁹⁵ Vgl. Mullenweg, 2008 [www-dokument] - WordPress 2.6

⁹⁶ Vgl. Mullenweg, 2008 [www-dokument] - WordPress 2.7

⁹⁷ Vgl. Mullenweg, 2008 [www-dokument] - WordPress 2.7

⁹⁸ Vgl. Mullenweg, 2008 [www-dokument] - WordPress 2.7

⁹⁹ Vgl. Mullenweg, 2008 [www-dokument] - WordPress 2.7

WordPress 2.8 (CHET BAKER): Am 11. Juni 2009 wurde die Version 2.8 veröffentlicht. Herausragend bei dieser Version ist hauptsächlich die Integration von THEMES über das WordPress Backend, in Analogie der PLUGIN-Einbindung von Version 2.7. Beim THEME- UND PLUGIN-EDITOR des Systems wurde das sogenannte ‚Syntaxhighlighting‘, Hervorhebung bestimmter Syntax-Elemente, eingeführt. Des Weiteren wurde die Schnelligkeit des Adminmenüs verbessert und die Datenbank-performance erhöht¹⁰⁰.

Neben den großen Veröffentlichungen wurden immer wieder kleine Teilversionen mit Verbesserungen und Fehlerkorrekturen herausgebracht. Gerade Teilversionen schlossen Sicherheitslücken, die bei Hauptversion noch nicht genügend ausgereift waren¹⁰¹.

Neben allen regulären veröffentlichten Versionen gibt es ebenfalls Nebenprojekte von WordPress. Diese stehen unter dem Namen WordPress μ oder BUDDY und werden je nach Projekt zu den regulären Erscheinungen von WordPress herausgebracht. WordPress μ ist eine Multiuser-Version des Systems und verbindet mehrere Blogs miteinander. So können beispielsweise durch WordPress μ viele Blogeinträge verwaltet und erstellt werden¹⁰². Zum ‚WordPress-Camp‘ am 30. Mai 2009 in San Francisco kündigte der Hauptverantwortliche, MATTHEW MULLENWEG an, dass ab der WordPress Version 3.0, die Systeme vereint werden sollen.¹⁰³

BUDDYPRESS ist dagegen eine Erweiterung von WordPress und WordPress μ . Ziel ist es, die Standardinstallation von WordPress μ für Communities oder Social Networks zu nutzen. BUDDYPRESS verknüpft die PLUGIN-Sammlung von WordPress μ und erweitert diese mit einzelnen Features, wie beispielsweise Privatnachrichten.¹⁰⁴

¹⁰⁰ Vgl. Mullenweg, 2009 [www-dokument] - WordPress 2.8 - 2.8 Release Jazzes THEMES and Widgets

¹⁰¹ Vgl. Alby, 2008a, S. 8

¹⁰² Siehe: <http://wpmu.de/>

¹⁰³ Vgl. Schmitz, 2009 [www-dokument] - WP und WPMU sollen vereint werden

¹⁰⁴ Siehe: <http://buddypress.de/>

4. WP-Backend und WCMS-Eigenschaften

4.1 Installation und Konfiguration

Die Installationsvoraussetzungen von WordPress auf einem Webserver beschränken sich im Wesentlichen auf zwei Hauptmerkmale. Der Server auf dem WordPress zum Einsatz kommt, sollte mindestens eine MySQL-Datenbank Version 4.0 besitzen und über die PHP Version 4.3 verfügen.¹⁰⁵

Für suchmaschinenfreundliche URLs bietet es sich zusätzlich an, dass Apache-Modul ‚mod_rewrite‘ zu installieren. Mit der ‚Rewrite-Engine‘ ist es möglich, auf Basis ‚regulärer Ausdrücke‘, URLs umzuschreiben und somit die Darstellung zu manipulieren.¹⁰⁶ Eine optionale Voraussetzung ist das ‚GD-Modul‘, welches vorrangig für die Generierung von automatischen Bildern verantwortlich ist. Gerade für Antispam-Maßnahmen eignet sich dieses Modul, um sogenannte CAPTCHAs zu erzeugen¹⁰⁷. Für die Pflege von MySQL-Datenbanken empfiehlt sich außerdem das Datenbankverwaltungswerkzeug PHPMYADMIN.

Die Installation von WordPress beginnt mit dem Herunterladen der aktuellen Version des Systems von der Betreiberplattform¹⁰⁸. Hier findet man in der Regel zwei unterschiedliche Dateiversionen im Zip-Format vor, wobei beide Dateien die offizielle Version des Systems beinhalten und sich lediglich in der Sprachauswahl voneinander unterscheiden. Aufgrund von Übersetzungsvorgängen kann es zu zeitlichen Verzögerungen bei Neuerscheinungen kommen, so dass zeitweise nur die englische Sprachversion zur Verfügung steht.

Nachdem die Zip-Datei heruntergeladen und entpackt wurde, werden die beinhalteten Verzeichnisse ‚wp-admin‘, ‚wp-content‘, ‚wp-includes‘ sowie alle root-Dateien auf den Webserver geladen. Hervorzuheben ist an dieser Stelle die Installationsdatei ‚wp-config-sample.php‘. In der Datei wird anschließend der Datenbankname, Hostname, Nutzernamen sowie das aktuelle Passwort der Datenbank definiert. Mit Hilfe eines einfachen Text- oder Webeditors, wie beispielsweise ADOBE DREAMWEAVER, werden die Daten konfiguriert und der Dateiname in ‚wp-config.php‘ geändert.¹⁰⁹

¹⁰⁵ Vgl. Mullenweg, 2010 [www-dokument] – requirements

¹⁰⁶ Vgl. Alby, 2007, S. 91

¹⁰⁷ Vgl. Alby, 2008a, S. 45

¹⁰⁸ Siehe: <http://wordpress-deutschland.org/download/>

¹⁰⁹ Vgl. Mullenweg, 2010 [www-dokument] – Installing_WordPress

```
01 // ** MySQL Einstellungen ** //
02 define(DB_NAME`, ,putyourdbnamehere`); // Definition von dem Namen der Datenbank
03 define(DB_USER`, ,usernamehere`); // Definition von MySQL-Datenbank-Benutzernamen.
04 define(DB_PASSWORD`, ,yourpasswordhere`); // Definition von MySQL-Passwort.
05 define(DB_HOST`, ,localhost`); // 99% keine Aenderung
06 define(DB_CHARSET`, ,utf8`); // Definition des Datenbankzeichensatz.
07 define(DB_COLLATE`, ,`);
08 define(AUTH_KEY`, ,put your unique phrase here`);
09 define(SECURE_AUTH_KEY`, ,put your unique phrase here`);
10 define(LOGGED_IN_KEY`, ,put your unique phrase here`); .
11 define(NONCE_KEY`, ,put your unique phrase here`);
12 $table_prefix = 'wp_'; // Nur Zahlen, Buchstaben und Unterstriche bitte!
13 // Definition welche Sprachdatei verwendet werden soll
14 define (WPLANG`, ,de_DE`);
15 if ( !defined(ABSPATH`) )
16     define(ABSPATH`, dirname(__FILE__) . ,/`);
17 require_once(ABSPATH . 'wp-settings.php');
```

Listing 4.1: Ausschnitt der Konfigurationsdatei ‚wp-config.php‘

Da WordPress ab der Version 2.6 die Eigenschaft besitzt, Beiträge und Seiten nicht zu überschreiben, sondern eine Backupdateien erzeugt, kann es zu vielen unnötigen Datenbankeinträgen kommen, wodurch sich die Performance des Systems verschlechtern kann. Aus diesem Grund bietet es sich an innerhalb der Konfigurationsdatei folgende Konstanten zu definieren¹¹⁰:

```
18 define(WP_POST_REVISIONS`, 3); // Gibt bei der wiederholten Speicherung nur 3
    automatischen Speicherung in die DB
```

Listing 4.2: Zusatz: Mit ‚automatischer Speicherung‘ in der Datei ‚wp-config.php‘

¹¹⁰ Vgl. Bültge, 2009 [www-dokument] – WordPress Konstanten nutzen

Sollte die automatische Intervallspeicherung nicht genutzt werden, so setzt man die Eigenschaft auf ‚false‘:

```
19 define(AUTOSAVE_INTERVAL, false); // Ohne automatische Speicherung
```

Listing 4.3: Zusatz: Ohne ‚automatischer Speicherung‘ in der Datei ‚wp-config.php‘

Nachdem die Konfigurationsdatei erstellt wurde, wird diese in das ‚_root‘ Verzeichnis der WordPress-Umgebung auf dem Webserver via FTP transferiert. Über den Browser kann nun die Webadresse inklusive der Datei ‚index.php‘ aufgerufen werden, auf welchem sich die WordPress-Installation befindet. Es folgt, die für WordPress bekannte „5 Minuten Installation“.¹¹¹



Abbildung 4.1: WordPress Installationsoberfläche

¹¹¹ Vgl. Mullenweg, 2010 [www-dokument] – Installing WordPress – Famous 5-Minute Install

Nachdem der Titel des Blogs sowie die E-Mailadresse bestimmt wurden (siehe Abbildung 4.1), wird mit ‚sendmail‘, falls auf dem Server konfiguriert, eine E-Mail an die angegebene Adresse übermittelt. Mit dieser Methode gelangt der Administrator, zu den Zugangsdaten und wird später über die angegebene Adresse informiert, ob neue Kommentare zu Artikeln verfasst wurden.¹¹²



Abbildung 4.2: Passwortvergabe innerhalb des Installationsvorgang

Die Administrationsoberfläche und den Login-Bereich von WordPress erreicht man unter der installierten Domain mit dem Anhang ‚wp-login.php‘. Nach Eingabe des Benutzernamens und Passworts wird man direkt in das Backend des Systems geleitet. Falls Benutzername oder Passwort fehlerhaft, oder nicht vergeben sind, folgt die Eingabemaske erneut mit dem Hinweis, dass Passwort oder Benutzername nicht übereinstimmen. Da diese Informationen in WordPress detailliert aufgeführt werden, kann ein potenzieller Angreifer die Informationen nutzen, um Hacking-Versuche zu optimieren. Hierdurch entsteht ein erhöhtes Sicherheitsrisiko, welche sich jedoch mit PLUGINS und Änderungen im Kern des Systems (CORE) beseitigen lässt¹¹³. Es sei erwähnt, dass Änderung im CORE von WordPress, Unstimmigkeiten bei Aktualisierungen zufolge haben können, weshalb es vermieden werden sollte, Programmdateien in den Verzeichnissen ‚wp-admin‘ und ‚wp-includes‘ zu modifizieren.¹¹⁴

¹¹² Vgl. Alby, 2008a, S. 48

¹¹³ Vgl. Bültge, 2008 [www-dokument] – WordPress Login Sicherheit - Secure WP

¹¹⁴ Vgl. o.A., 2008, [www-dokument]

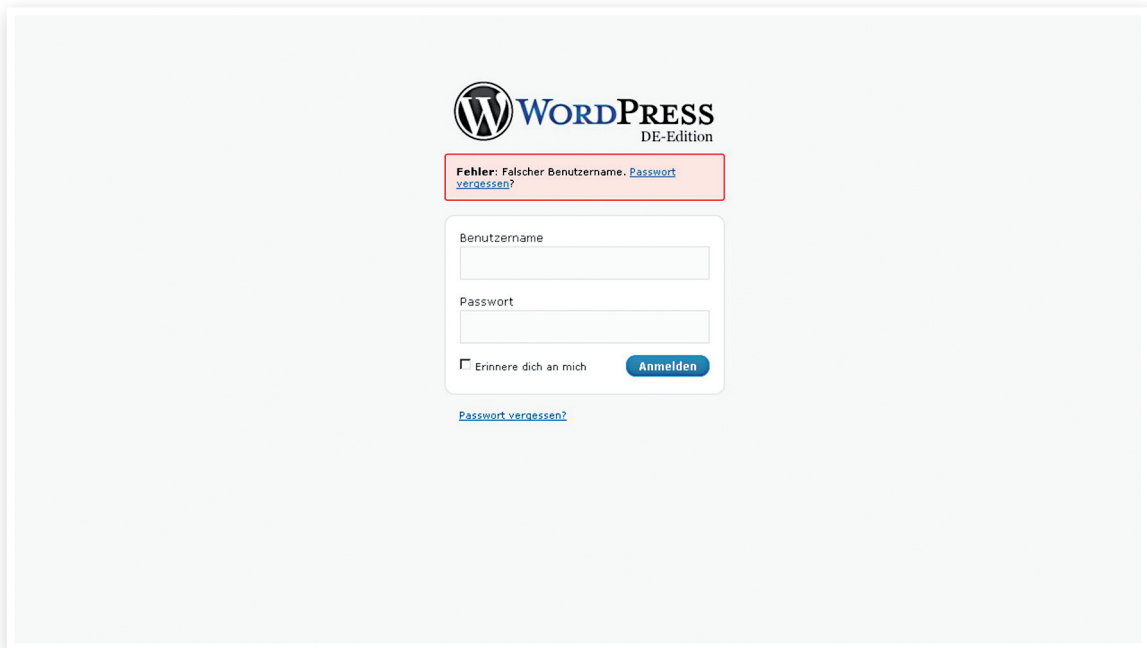


Abbildung 4.3: Login-Bereich von WORDPRESS mit dem Hinweis auf falschen Benutzername

4.2 Administrationsoberfläche von WORDPRESS

Im Folgenden wird der grundlegende Aufbau und die Strukturen der Administrationsoberfläche von WORDPRESS anhand des persönlichen Nutzungsverhaltens erläutert. Die Untersuchung, in wie weit typische Systemfeatures von Web Content Management Systemen im Blogsystem WORDPRESS zu finden sind, steht hierbei im Vordergrund.

4.2.1 DASHBOARD

Der Nutzer gelangt nach erfolgreichem Login auf eine Gesamtübersicht des Systems – genannt DASHBOARD. Hierbei bekommt man einen groben Überblick über die Features und die bereits vorhandenen Inhalte.

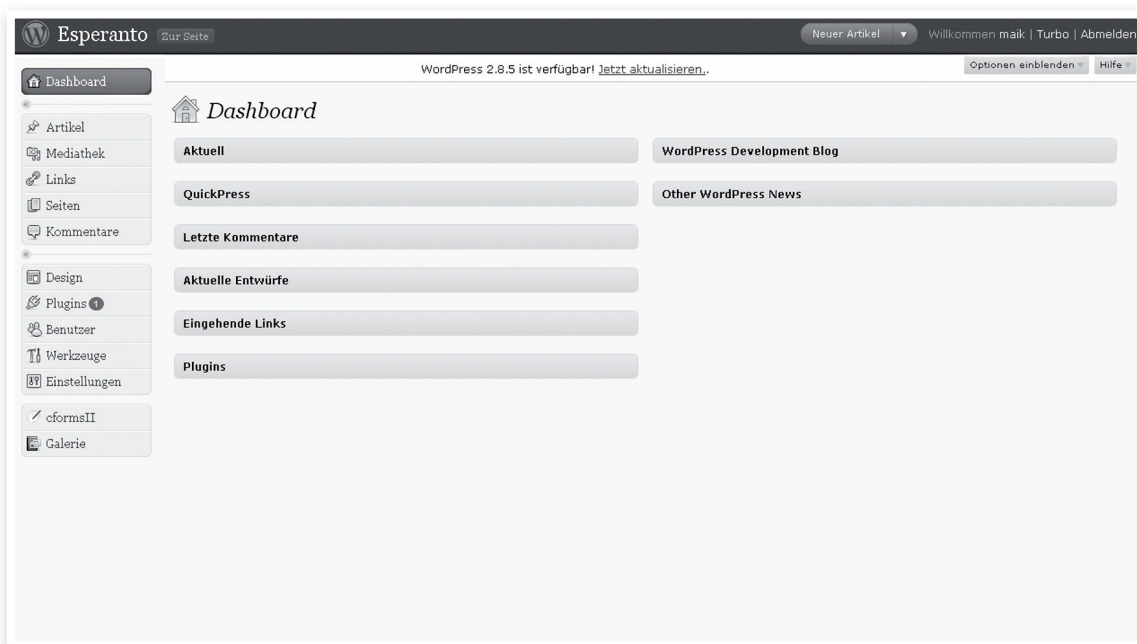


Abbildung 4.4: DASHBOARD von WORDPRESS

Das DASHBOARD, wie auch die restliche Administrationsoberfläche von WORDPRESS, unterscheidet sich in drei Kernbereiche. Am oberen Rand (Header), findet man den Titel des Systems sowie eine Schnellnavigation für die Erstellung neuer ARTIKEL, SEITEN, ENTWÜRFE und KOMMENTARE. Unterhalb des ‚Headers‘ integrierten die Entwickler eine zusätzliche Ebene, die über Neuerscheinungen informiert oder Hilfestellungen zu speziellen Themengebieten leistet. Am linken Rand der Administrationsoberfläche befindet sich die Hauptnavigation, sowie mittig der Contentbereich.¹¹⁵

Im Contentbereich des DASHBOARDS erhält der Nutzer unter dem Button AKTUELL eine Zusammenfassung über bestehende Inhalte. Es wird verdeutlicht, ob ARTIKEL verfasst wurden und ob diese TAGS oder KATEGORIEN enthalten, sowie die Anzahl der veröffentlichten SEITEN im Frontend. Darüber hinaus bekommt man Informationen, welche Layoutvorlage und aktuelle WORDPRESS-Version verwendet wird.¹¹⁶

¹¹⁵ Vgl. Belik, 2010, S. 190f.

¹¹⁶ Vgl. Bültge, 2009, S. 71

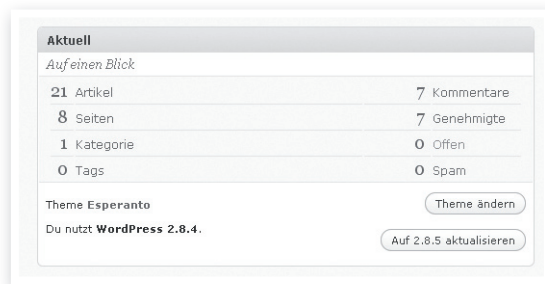


Abbildung 4.5: Menüpunkt AKTUELL im DASHBOARD

Der ab Version 2.7¹¹⁷ integrierte QUICKPRESS-EDITOR bietet die Möglichkeit, aus dem DASHBOARD, ARTIKEL zu verfassen und diese zu veröffentlichen. Das Grundprinzip von QUICKPRESS ist eine vereinfachte und minimale Schnittstelle, ARTIKEL zu erstellen. Im Gegensatz zum Hauptmenüpunkt ARTIKEL des Backends, verwendet QUICKPRESS nur die nötigsten Funktionen, um ARTIKEL zu erstellen. Im WCMS-Bereich überzeugt diese Funktion vor allem durch ihre einfache Bedienung und ist für das schnelle Publizieren effektiv.¹¹⁸

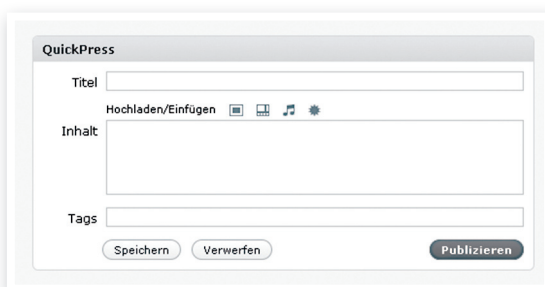


Abbildung 4.6: Menüpunkt QUICKPRESS im DASHBOARD

Die Punkte KOMMENTARE und ENTWÜRFE zeigen im DASHBOARD in übersichtlicher Weise die letzten KOMMENTARE und aktuelle ENTWÜRFE des Systems. Im Punkt KOMMENTARE werden dabei die letzten fünf KOMMENTARE von ARTIKELN oder Gästebucheinträgen zusammengefasst. Hier können Inhalte von KOMMENTAREN überblickt werden und diese bei nicht gefallen zurückgewiesen, bearbeitet, gelöscht oder als Spam markiert werden. Ebenfalls besteht die Gelegenheit, direkt zum Haupt-

¹¹⁷ Vgl. Mullenweg, 2008 [www-dokument] - WORDPRESS 2.7

¹¹⁸ Vgl. Bültge, 2009, S. 71

menüpunkt KOMMENTARE zu wechseln. Der Anstrich ENTWÜRFE dient zur Visualisierung der letzten gespeicherten ENTWÜRFE. Gerade Benutzergruppen mit eingeschränkten Rechten (z.B. Redakteure), können vom Autor verfasste ENTWÜRFE, schneller einsehen und veröffentlichen.

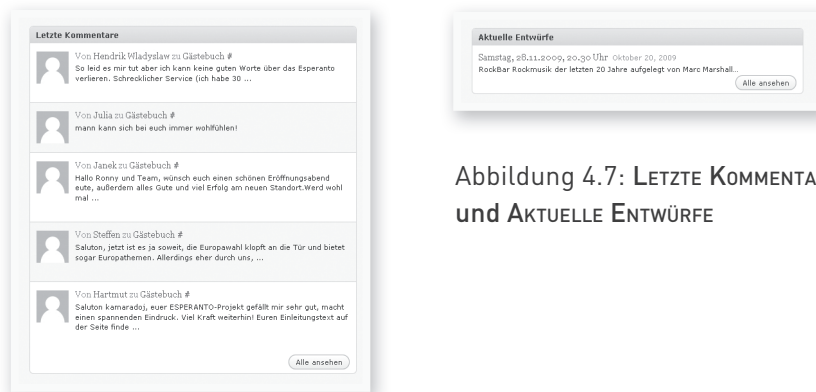


Abbildung 4.7: LETZTE KOMMENTARE und AKTUELLE ENTWÜRFE

Die Menüpunkte im DASHBOARD, EINGEHENDE LINKS, PLUGINS, WORDPRESS DEVELOPMENT BLOG und WORDPRESS NEWS werden über externe Server in die Administrationsoberfläche von WordPress implementiert. Innerhalb der EINGEHENDEN LINKS wird über ein WIDGET¹¹⁹, die Suchmaschine GOOGLE BLOGS¹²⁰ durchsucht, ob andere Blogs oder WORDPRESS-Systeme auf die persönliche Seite verlinken. Ist dies der Fall, werden die Treffer chronologisch dargestellt und geordnet.¹²¹

Ähnlich verhält es sich bei dem Menüpunkt PLUGIN. Sobald neue WORDPRESS-Erweiterungen auf www.wordpress.org veröffentlicht werden, erscheinen die drei letzten veröffentlichten PLUGINS im Menü. Da WORDPRESS in der Nutzung als Web Content Management System auf Erweiterungen angewiesen ist, stellt dieser Menüpunkt vielseitige Informationen für den Einsatz dar. Das Menü zeigt weiterhin, Aktualisierungen sowie Fehlerbehebungen von Erweiterungsmodulen und ermöglicht die direkte Installation einer aktuelleren Version aus dem DASHBOARD.¹²²

Unter den Punkten WORDPRESS DEVELOPMENT und OTHER WORDPRESS NEWS werden via XML-Schnittstelle RSS FEEDS importiert¹²³. Dabei ergeben sich noch umfangreichere und detailliertere Neuigkeiten aus dem Bereich WORDPRESS-Entwicklung und Veröffentlichung. Das Besondere

119 Siehe: http://codex.wordpress.org/WordPress_Widgets

120 Siehe: <http://blogsearch.google.com/>

121 Vgl. Mullenweg, 2010 [www-dokument] – Administration Panels Dashboard

122 Vgl. Mullenweg, 2010 [www-dokument] – Dashboard Updates

123 Vgl. Belik, 2010, S.180ff.

an dieser Funktion ist, dass FEEDS nach persönlichen Interessen angepasst und konfiguriert werden können. Möchte der Nutzer anstelle der WORDPRESS DEVELOPMENT NEWS, aktuelle Tagesereignisse importieren, so kann mit Hilfe des WIDGETS auch beispielsweise der FEED von tagesschau.de eingebunden werden. Somit ist die Nutzung von WORDPRESS auch als sekundärer FEEDREADER¹²⁴ möglich, und kann nach persönlichen Vorlieben angepasst werden.¹²⁵

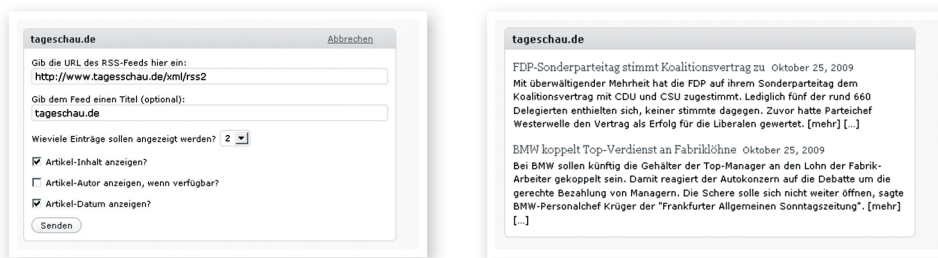


Abbildung 4.8: Konfiguration des FEEDREADER mit RSS-FEED von tagesschau.de

4.2.2 ARTIKEL

Das Bedienfeld ARTIKEL ist das Herzstück von WORDPRESS. Über den Menüpunkt in der Administrationsoberfläche ARTIKEL werden Inhalte erstellt, publiziert, kategorisiert, archiviert und editiert. Sämtliche Inhalte, inklusive der Einbindung von Medientypen, können im Menüpunkt ARTIKEL vorgenommen und verwaltet werden. Falls eine Überzahl an Beiträgen dominiert, bietet es sich aus Gründen der Übersichtlichkeit an, eine Kurzfassung des Textes mit prägnanten Überschriften zu wählen. Des Weiteren ist die Darstellung der gelisteten Beiträge über eine Seitennavigation wählbar und nach Kategorien sowie Erstellungsdatum zu filtern.

Innerhalb der Gesamtübersicht können einzelne Beiträge direkt gelöscht oder bearbeitet werden. Ein sehr vorteilhaftes und zeitsparendes Verfahren, ergibt sich bei einer Mehrfachauswahl von Beiträgen. Hierbei ist es denkbar, mehrere Beiträge unter der erstellten Auswahl zu bearbeiten. Es können so ausgewählte Beiträge neuen Kategorien zugewiesen und Änderungen an TAGS, AUTOR, VERÖFFENTLICHUNGSSTATUS oder KOMMENTAREN vorgenommen werden. Diese Funktion steht in Analogie zur Hauptbearbeitungsfunktion (ERSTELLEN und BEARBEITEN) von ARTIKELN, welche einen wesentlich größeren Bearbeitungsumfang erlaubt.

¹²⁴ Siehe <http://www.feedreader.com/>

¹²⁵ Vgl. Belik, 2010, S.180f.

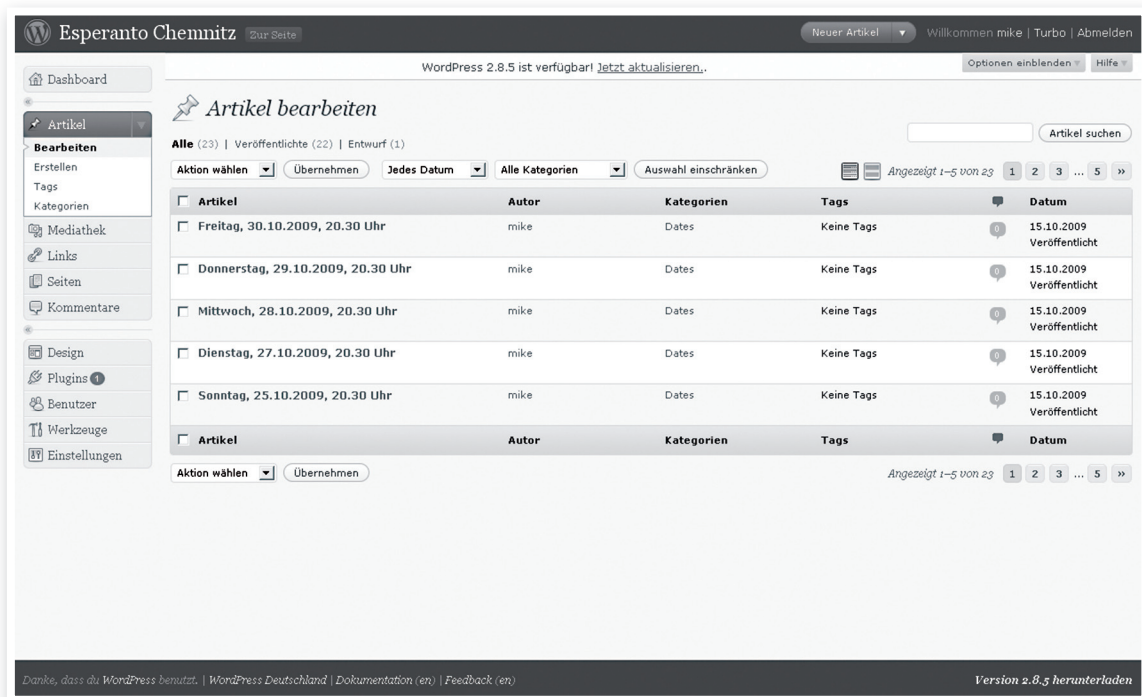


Abbildung 4.9: Artikelübersicht von WordPress

4.2.2.1 ARTIKEL ERSTELLEN UND BEARBEITEN

Da die Menüpunkte ERSTELLEN und BEARBEITEN innerhalb der ARTIKEL sich nicht wesentlich voneinander unterscheiden, werden die Reiter im Abschnitt gemeinsam diskutiert. Beide besitzen eine Vielzahl an Funktionen, die bei der Erstellung von Inhalten innerhalb eines WCMS relevant sind. Um die Eigenschaften genau zu definieren, werden alle Elemente mit Hilfe des folgenden Screenshootes inklusive einer Legende erörtert. Dabei ist die Anordnung der Zahlen 3 bis 10 zufällig gewählt und entspricht keiner optimalen Arbeitsumgebung. Die gekennzeichneten Bereiche können mittels WordPress interner AJAX- und JQuery¹²⁶ Frameworks beliebig verschoben werden.

126 Siehe: www.jquery.com/

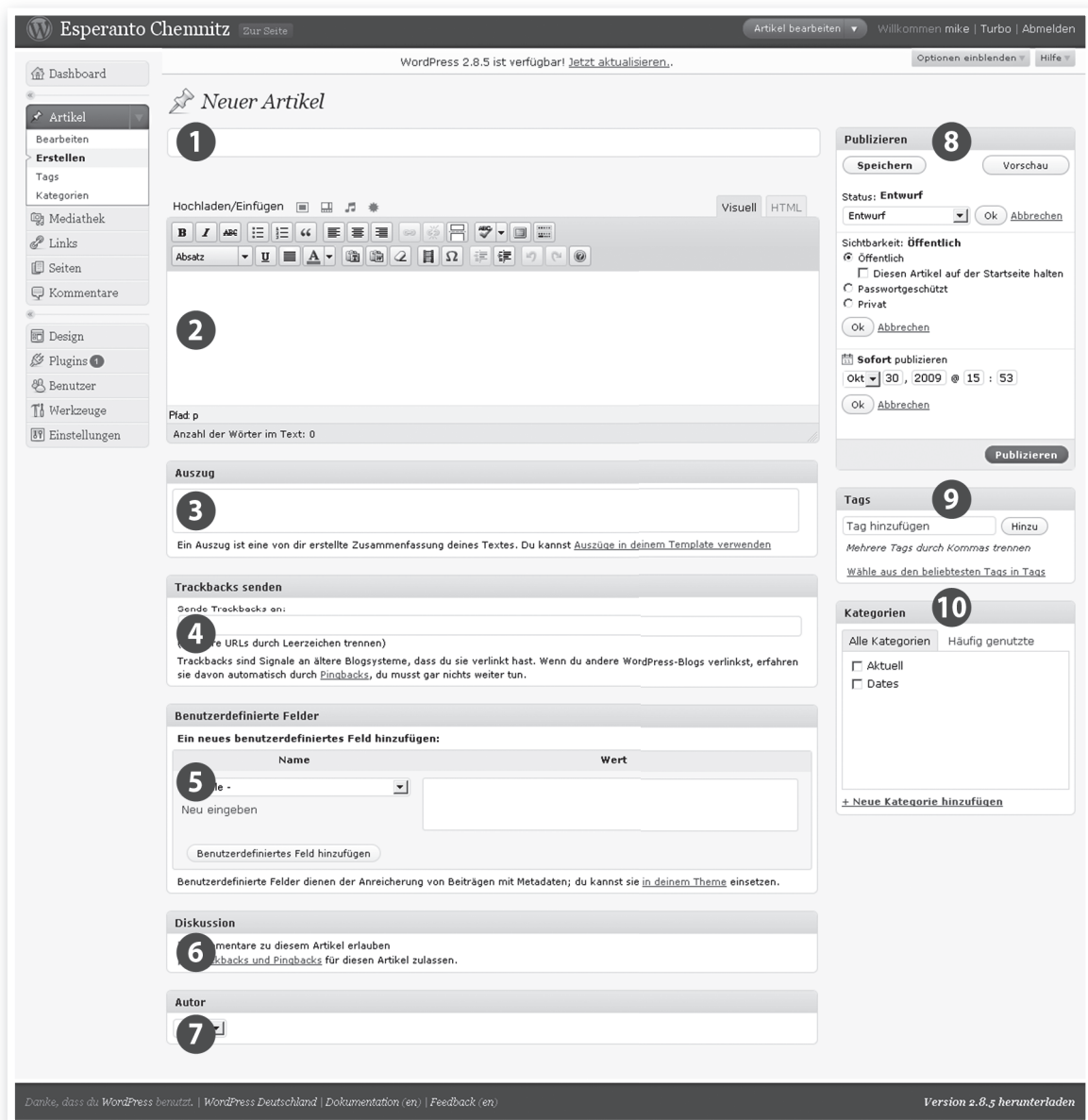


Abbildung 4.10: Erstellung von ARTIKELN innerhalb des WORDPRESS-Backends

1 TITEL: Der TITEL wird sowohl als Überschrift im Backend für die Gesamtdarstellung der Beiträge genutzt, als auch für die Darstellung der Titel im Frontend. In den meisten THEMES wird der TITEL im Frontend als H2-HTML-Tag definiert, sodass der Beitrag eine höhere Positionierung bei Suchmaschinen erhalten kann¹²⁷. Auch in FEEDS kommt der TITEL als strukturierendes Element zum Einsatz¹²⁸. Im WORDPRESS DEFAULT-THEME wird der TITEL folgendermaßen deklariert:

```
01 <h2><a href="<?php the_permalink() ?>" rel="bookmark" title="
<?php printf(__('Permanent Link to %s', 'kubrick'), the_title_attribute('echo=0'));?>">
<?php the_title(); ?></a></h2>
```

Listing 4.3: Der TITEL im WORDPRESS DEFAULT-THEME

Der TITEL wird dabei durch den TEMPLATE TAG `'the_title()'` im Frontend ausgegeben. Dieser befindet sich im DEFAULT-THEME innerhalb eines H2 und a-Tags. Der TEMPLATE TAG `'the_title'` darf nur innerhalb des WORDPRESS LOOPS aufgerufen werden und kann die Parameter `'before'`, `'after'`, `'display'` enthalten.¹²⁹ Genauere Definitionen und Anwendungen von WORDPRESS spezifischen TEMPLATE TAGS, CONDITIONAL TAGS und die Besonderheiten des LOOPS werden im Kapitel 5 eruiert.

2 EDITOR: Die Erstellung aller ARTIKEL, ENTWÜRFE und SEITEN erfolgt mit dem RICHTEXT-EDITOR. Seit der Version 2.1 ist dabei das Umschalten zwischen einer visuellen und einer Code-Ansicht, ohne Zwischenspeicherung, möglich¹³⁰. Der RICHTEXT-EDITOR ist keine WORDPRESS eigene Software, sondern stammt vom Open Source-Projekt TINYMCE¹³¹. Im Standardmodus kann über den Button WERKZEUGLISTE ANZEIGEN/VERSTECKEN eine zusätzliche Auswahl an Funktionsfeldern dem Editor hinzugefügt werden. Über diese Funktion werden beispielsweise verschiedene Formatvorlagen wie Überschriften oder Absätze mitgeteilt, die über CSS formatiert werden.

¹²⁷ Vgl. Siefken, 2008, S. 43

¹²⁸ Vgl. Bültge, 2007, S. 56

¹²⁹ Vgl. Mullenweg, 2010 [www-dokument] – TEMPLATE TAGS/the title

¹³⁰ Vgl. Bültge, 2007, S. 56 ff.

¹³¹ Siehe: <http://tinymce.moxiecode.com/>

Weitere Features des TINYMCE-EDITORS sind zum Beispiel das Einfügen von Worddokumenten, die Einbettung von Medien, sowie das Einrücken und Formatieren von Texten. Darüber hinaus kann der Editor mit dem PLUGIN TINYMCE ADVANCED 2 erweitert werden. Vor allem CSS können mit der Erweiterung explizit ausgewählt und angewandt werden.¹³²

Ein wichtiges WCMS-Feature wurde ab der WORDPRESS Version 2.1, dem Editor implementiert. Falls es zu längeren Bearbeitungsphasen an Inhalten, oder durch den Browser verursachten Abstürzen kommt, erstellt WORDPRESS eine automatische Sicherung. Diese ist wie im Kapitel 4.1 beschrieben, mit Hilfe von Konstanten in der Datei ‚wp-config.php‘ definiert. Sämtliche Sicherungen werden nach Ablauf des Intervalls in den ENTWÜRFEN gespeichert und können von da aus, erneut bearbeitet und veröffentlicht werden.

Ein besonderes WCMS-Feature von WORDPRESS ist der spezifische ‚more-Tag‘ innerhalb des Editors. Dieser TAG erlaubt dem Autor, Beiträge als Teaser zu veröffentlichen, wobei der Leser durch den Vorspann einen kurzen Eindruck vom jeweiligen Thema bekommt. Gerade bei Web Content Management Systemen ist diese Funktionalität von enormen Vorteil, da textliche Inhalte platzsparend im Frontend abgebildet werden. Die QUICKTAGS ‚more-Tag‘ und ‚nextpage-Tag‘ werden an der gewünschten Stelle mit ‚<!-- more-->‘ oder mit ‚<!--nextpages -->‘ in den Text des Editors integriert. Dabei verweist das ‚nextpage-Tag‘ bei längeren Texten auf die Folgeseiten des Beitrages. Das ‚more-Tag‘ dagegen, kann direkt über den Editor unter dem definierten Symbol oder mit der Tastenkombination Alt+Shift+T (Firefox) in den Text eingefügt werden.¹³³

3 AUSZÜGE: Ein weiteres bedeutsames WCMS-Feature von WORDPRESS sind AUSZÜGE. Fügt man kurze oder zusammenfassende Inhalte eines Beitrags in das Feld ein, werden diese im Backend in der Artikelgesamtübersicht gelistet. AUSZÜGE dienen vor allem der Strukturierung und Eingrenzung des Themenbereichs. Im Backend sind damit einzelne Beiträge in der Artikelübersicht schneller erreichbar und verweisen auf die jeweiligen Themen. Falls beispielsweise der Titel des Beitrags nicht explizit den Inhalt des Beitrages wiedergibt und unterschiedliche Autoren, Beiträge editieren, kann es schnell zu Verwechslungen kommen. Um ein solches Ereignis zu vermeiden, werden innerhalb der AUSZÜGE die Kerninhalte des Beitrages beschrieben und sichtbar im Backend aufgezeigt.¹³⁴

AUSZÜGE können auch als Zusammenfassung von Beiträgen im Frontend fungieren. Enthält ein Beitrag einen AUSZUG, so wird dieser mit maximal 120 Zeichen im Frontend veröffentlicht.

¹³² Vgl. Ozz, 2009 [www-dokument]

¹³³ Vgl. Bültge, 2009, S 56 ff.

¹³⁴ Vgl. Bültge, 2009, S 56 ff.

Voraussetzung ist jedoch die Verwendung des `TEMPLATE TAGS` `,the_excerpt()'` innerhalb des `WORDPRESS-LOOPS`.¹³⁵ Der Nachteil von `'the_excerpt'` ist allerdings, dass der `TEMPLATE TAG` keine Parameter verwenden darf. Sollte beispielsweise ein Auszug länger als 120 Zeichen sein, wird dieser durch die Zeichen `,[...]` getrennt. Des Weiteren ignoriert die Funktion sämtliche Formatierung innerhalb des Textes. In der aktuellen `WORDPRESS` Version 2.9 kann mit Hilfe von Filtern der `TEMPLATE TAG` `'the_excerpt'` manipuliert werden. Dabei wird in die Datei `,functions.php'` des genutzten `THEMES` folgender Code hinzugefügt¹³⁶:

```
01  <?php
// ändern der excerpt Länge auf 50 Wörter
02  function new_excerpt_length($length) {
03      return 50;
04  }
05  add_filter('excerpt_length', 'new_excerpt_length');
// ändern der excerpt „more“ Zeichenkette - gibt den Wert „mehr erfahren...“ zurueck
06  function new_excerpt_more($more) {
07      return 'mehr erfahren...';
08  }
09  add_filter('excerpt_more', 'new_excerpt_more');
10  ?>
```

Listing 4.4: Erstellen von Auszügen mit dem `TEMPLATE TAG` `'the_excerpt'`

4 TRACKBACKS SENDEN: In diesem Backend-Modul können `TRACKBACK-URLs` manuell eingegeben werden. In der Regel werden für Blogs, `TRACKBACKs` in Form von Kommentaren automatisch erfasst. `TRACKBACKs` weisen auf ein anderes Blog hin, welches ähnliche Themen publiziert. Da die `Trackbackfunktion` innerhalb von `Web Content Management Systemen` nicht von großer Relevanz ist, wird sie hier nicht weiter erläutert.¹³⁷

5 BENUTZERDEFINIERT FELDER: Die `BENUTZERDEFINIERT FELDER` auch `CUSTOM FIELDS` genannt, sind mächtige `WCMS-Instrumentarien` von `WORDPRESS`. Mit Hilfe dieser Felder können Metainforma-

¹³⁵ Vgl. Mullenweg, 2010 [www-dokument] – `TEMPLATE TAGS/the_excerpt`

¹³⁶ Vgl. Oeser, 2009 [www-dokument]

¹³⁷ Vgl. Thon-Soun, 2006 [www-dokument]

tionen zum jeweiligen Beitrag oder Seite vergeben werden. Die Inhalte der Felder werden dabei in der Datenbanktabelle ‚wp_postmeta‘ gesichert. Jedes vergebene Feld wird ein Schlüssel (key) und ein eindeutiger Wert (value) zugeordnet. Der ‚Key‘ lässt sich dabei mehrfach vergeben und der ‚Value‘ ist mit einer eindeutigen ‚ID‘ deklariert. Es lassen sich unendlich viele solcher Attribute übergeben, die später innerhalb des LOOPS mit TEMPLATE TAGS ausgelesen werden. Für die Ausgabe der Werte im Frontend stehen die TEMPLATE TAGS ‚the_meta()‘ und ‚get_post_meta()‘ zur Verfügung.¹³⁸

Ein Beispiel für die Verwendung von BENUTZERDEFINIERTEN FELDERN ist die Einbindung von einem Werbefbanner vor, nach oder innerhalb eines definierten ARTIKELS. Im Backend wird beispielsweise in den BENUTZERDEFINIERTEN FELDERN der Name ‚topbanner_on‘ und der Wert=1 vergeben. Nach Sicherung und Veröffentlichung des ARTIKELS, soll der Werbefbanner (topbanner_on) oberhalb des geschriebenen Beitrags im Frontend erscheinen. Hierfür muss im WORDPRESS TEMPLATE-Verzeichnis die ‚index.php‘ oder ‚single.php‘ angepasst werden. Für das beschriebene Beispiel soll der Banner in der Gesamtdarstellung aller gelisteten Beiträge erscheinen und wird deshalb, in die Datei ‚index.php‘ des THEMES eingebunden.

```
01 // Wenn Beitrag mit definierten Feld vorhanden ist, dann gibt den Banner aus
02 <?php if(get_post_meta($post->ID, 'topbanner_on', true)) : ?>
03     
04 <?php endif; ?>
```

Listing 4.5: Verwendung von BENUTZERDEFINIERTEN FELDERN

BENUTZERDEFINIERT FELD haben jedoch weitere Möglichkeiten. So können auch Musiktitel, Tabellen, Flashbanner, Formulare oder Umfragen ausgegeben und definiert werden.¹³⁹

6 DISKUSSION: Über die Box DISKUSSION werden KOMMENTARE, TRACKBACKS und PINGBACKS gesteuert. Enthalten diese Felder kein Häkchen, so wird die Auswahl eingeschränkt oder deaktiviert.¹⁴⁰ Standardmäßig sind PINGBACKS und KOMMENTARE aktiv. Alle getroffenen Einstellungen sind ausschließlich für den jeweiligen Artikel bestimmt und gelten nicht global.

¹³⁸ Vgl. Mullenweg, 2010 [www-dokument] – Custom Fields

¹³⁹ Vgl. Mullenweg, 2010 [www-dokument] – Custom Fields

¹⁴⁰ Vgl. Bültge, 2009, S. 60

7 AUTOR: Über eine `SELECTBOX` können in diesem Bereich alle bestehenden Autoren des Systems zum Erstellen von Beiträgen ausgewählt werden.

8 PUBLIZIEREN: Im Abschnitt `PUBLIZIEREN` wird sowohl der Status eines Artikels als auch die zeitliche Veröffentlichung abgebildet. Das Modul wird dabei in drei unterschiedliche Bereiche geteilt. Im ersten Bereich befindet sich der Veröffentlichungsstatus, wobei dieser Status drei Zustände einnehmen kann:

VERÖFFENTLICHT: Der Artikel wurde bereits veröffentlicht und ist im Frontend sichtbar.

AUSSTEHENDER REVIEW: Der Artikel wurde von einem Nutzer verfasst, der keine Veröffentlichungsrechte besitzt. Der Artikel befindet sich in der Warteschleife und muss vom Autor oder Administrator freigegeben werden.

ENTWURF: Der Artikel ist noch nicht veröffentlicht und muss eventuell bearbeitet werden.

Die zweite Gliederungsstufe des Abschnittes `PUBLIZIEREN` trägt die Bezeichnung `SICHTBARKEIT`. Hier wird unter anderem festgelegt, ob Artikel `ÖFFENTLICH SICHTBAR`, `PASSWORTGESCHÜTZT` oder `PRIVAT` sind.

ÖFFENTLICHE ARTIKEL: Es werden alle Artikel im Frontend dargestellt und sind für jeden Nutzer sichtbar. Eine Besonderheit ist jedoch ein Zusatzstatus dieses Buttons. Es kann durch Bestätigen der sogenannten `STICKY FUNKTION` der `ARTIKEL` immer an erster Stelle abgebildet werden. Das heißt, wenn beispielsweise dieser Punkt aktiv ist, werden Beiträge immer an oberster Stelle innerhalb der `ARTIKEL` veröffentlicht und fallen aus der chronologischen Wiedergabe heraus.¹⁴¹

PASSWORTGESCHÜTZT: Alle veröffentlichten Artikel können gleichzeitig mit einem Passwort geschützt werden. Falls ein Passwort in der Administrationsoberfläche festgelegt wurde, wird der Nutzer im Frontend aufmerksam gemacht, dass der jeweilige Artikel geschützt ist und der Zugang nur durch ein Passwort ermöglicht wird.

PRIVAT: Hier wird festgelegt, ob ein Beitrag direkt veröffentlicht oder der Status `PRIVAT` aktiviert ist. Die Funktion ist vorrangig als Vorschaufunktion für Autoren angedacht. Wenn der Status `PRIVAT` selektiert wird, dann ist dieser nicht für außenstehende Leser sichtbar, sondern ausschließlich für den Autor.¹⁴²

¹⁴¹ Vgl. Bültge, 2008 [www-dokument] – Sticky-Funktionalität ab WordPress 2.7

¹⁴² Vgl. Alby, 2008a, S. 70

Im dritten Bereich des Abschnittes PUBLIZIEREN wird der ZEITSTEMPEL definiert. Innerhalb des ZEITSTEMPELS ist das Veröffentlichungsdatum und die Zeitperiode bestimmbar. Wird diese Funktionalität aktiviert und ein entsprechendes Veröffentlichungsdatum angegeben, so erscheint der Artikel erst zum definierten Zeitpunkt im Frontend. Die Veröffentlichungszeiträume können allerdings auch nachträglich geändert werden. Sollen beispielsweise Inhalte aus einer statischen Website übernommen werden, können mit dieser Funktion auch die „realen Zeiträume“ wiederhergestellt werden.¹⁴³ Die Ausgabe der Zeit im Frontend funktioniert in diesem Zusammenhang mit dem TEMPLATE TAG (the_date). Ein kleines Beispiel soll diese Funktion verdeutlichen: Im Frontend wird unterhalb des Artikels das Veröffentlichungsdatum in einen <p> TAG inklusive der KLASSE ‚dates‘ ausgegeben. Hierzu muss lediglich in der Datei ‚index.php‘ des THEMES folgender Code eingebunden werden:

```
01 <?php the_date(,Y-m-d', ,<p class="dates">', ,</p>'); ?>
02 // ausgabe im frontend des datums: Jahr- Monat- Tag innerhalb der eines p-tag mit
    klasse
```

Listing 4.6: Verwendung des TEMPLATE TAGS ‚the_date‘

9 **TAG:** In diesem Abschnitt können verschiedene Schlagwörter, sogenannte TAGS, eingetragen werden, die den Inhalt des ARTIKELS beschreiben. Dies kann beispielsweise in Form von Wörtern, die im Beitrag vorkommen, oder mit Hilfe von Synonymen geschehen. Der Hauptvorteil von TAGS ist, dass eine unbegrenzte Anzahl an Wörtern und Begriffen den Inhalt des Beitrags umschreibt und kategorisiert. Alle spezifischen TAGS können in diesem Feld eingetragen und mehrere Wörter durch Kommas separiert werden.

10 Das letzte Modul welches standardmäßig ohne Erweiterung in der Artikeloberfläche integriert ist, wird als KATEGORIEN bezeichnet. Hier können alle verfassten Beiträge nach Rubriken sortiert und geordnet werden. KATEGORIEN können nacheinander ausgebaut und in bestimmte Ober- oder Unterkategorien unterteilt werden.

Für die Verwaltung der KATEGORIEN gibt es in Analogie zu den TAGS einen eigenen Reiter innerhalb des globalen Moduls ARTIKEL. Hier werden alle KATEGORIEN verwaltet, die in WORDPRESS Verwendung finden. Innerhalb der Artikelerstellung können Beiträge beliebig viele Kategorien enthalten.¹⁴⁴

¹⁴³ Vgl. Bültge, 2009, S. 62

¹⁴⁴ Vgl. Bültge, 2009, S. 60

4.2.2.2 TAGS und KATEGORIEN innerhalb der ARTIKEL

In dem Menüpunkt ARTIKEL befinden sich neben BEARBEITEN und ERSTELLEN zwei weitere Reiter, die nur kurz erwähnt werden sollen, da diese teilweise im vorherigen Abschnitt vorgestellt wurden. Die beiden Reiter definieren im Einzelnen TAGS oder KATEGORIEN und sind von ihrer Aufbauarchitektur relativ ähnlich. Die beiden Arten verwalten und beschreiben alle erzeugten KATEGORIEN und TAGS in einer Gesamtübersicht. Sowohl das LÖSCHEN als auch die BEARBEITUNG ist in diesem Segment möglich. Obwohl sich KATEGORIEN und TAGS scheinbar wenig voneinander unterscheiden, gibt es wesentliche Merkmale, welche sie voneinander trennen. Das Ordnungsprinzip von KATEGORIEN erfolgt immer nach einer Taxonomie. Demgegenüber geschieht die Vergabe von TAGS nach einzelnen Schlagwörtern, die nicht einer bestimmten Taxonomie unterliegen.¹⁴⁵

4.2.3 MEDIATHEK

Die MEDIATHEK ist ein großes Verwaltungsmodul für alle in WORDPRESS eingebundenen Medien. In der MEDIATHEK können sowohl Bilder, Galerien, Audiodateien als auch Videosequenzen verwaltet und organisiert werden. Die hochgeladenen Medien werden dabei nach ihrer Erstellung chronologisch geordnet, wobei die Möglichkeit besteht, die MEDIATHEK beim Verfassen von ARTIKELN und SEITEN aufzurufen und deren Inhalte zu integrieren.¹⁴⁶



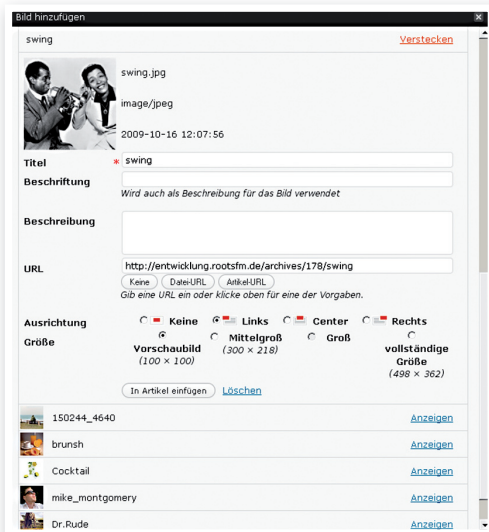
Abbildung 4.11: Einfügen von Medieninhalten

- 1 **BILDER HINZUFÜGEN:** Öffnet innerhalb von ARTIKELN oder SEITEN die MEDIATHEK und die Kategorie Bilder. Dabei unterstützt die MEDIATHEK im Standardmodus die webüblichen Bildformate .jpg, .gif und .png.
- 2 **VIDEO HINZUFÜGEN:** Öffnet innerhalb von ARTIKELN oder SEITEN die MEDIATHEK und die Kategorie Videos.
- 3 **AUDIO HINZUFÜGEN:** Öffnet innerhalb von ARTIKELN oder SEITEN die MEDIATHEK und die Kategorie Audio.

¹⁴⁵ Vgl. Alby, 2008a, S. 20

¹⁴⁶ Vgl. Wetzlmayr, 2008 [www-dokument]

4 DATEIEN HINZUFÜGEN: Öffnet innerhalb von ARTIKELN oder SEITEN die MEDIATHEK und die Kategorie Dateien. Es können in diesem Feld beispielsweise pdf-Dokumente oder swf-Dateien hochgeladen werden.



Bei der Einbindung von Medien über die MEDIATHEK können die Inhalte unter dem Bedienfeld AUSRICHTEN positioniert werden. Es besteht des Weiteren die Option, Bildern beispielsweise eine bestimmte Größe zuzuweisen. Leider ist nach eigenen Untersuchungen diese Einbindung eher suboptimal, da es schnell zu Ungenauigkeiten bei der Platzierung kommen kann. Eine modulare Platzierung von Medien und Inhalten, wie zum Beispiel beim WCMS TYPOLIGHT, wäre eine elegantere und sichere Lösung.¹⁴⁷

Abbildung 4.12: BILD HINZUFÜGEN in der MEDIATHEK

4.2.4 LINKS

Unter dem Hauptmenüpunkt LINKS findet man den sogenannten BLOGROLL des Systems. In diesem Bereich werden alle Links aufgeführt, die man eventuell den Lesern empfiehlt oder welche nützlich für den Inhalt der eigenen Seite sein könnten. Üblicherweise werden alle Links im Frontend auf einer separaten Seite oder in der SIDEBAR angezeigt. Die Verlinkung der Seite ist gerade für Suchmaschinenoptimierungsprozesse von hoher Relevanz¹⁴⁸. Das Ergebnis bei Suchmaschinen ist umso höher, je mehr Blogs- oder Seiten mit ähnlichen Inhalten, die eigene Seite verlinken.¹⁴⁹

4.2.4.1 LINKS BEARBEITEN und LINKS HINZUFÜGEN

In dem Untermenüpunkt LINKS BEARBEITEN befindet sich ein ähnlicher Aufbau und Strukturierung wie bei dem Unterpunkt ARTIKEL BEARBEITEN. Der Nutzer sieht eine Gesamtdarstellung aller vorhandenen Links, welche man aus diesem Dialog editieren oder löschen kann.

¹⁴⁷ Feyer, 2010 [www-dokument] - Typolight Open Source CMS Funktionen

¹⁴⁸ Jerkovic, 2010, S. 241 ff.

¹⁴⁹ Adler, 2009, S. 47

Im Untermenüpunkt **LINKS HINZUFÜGEN** werden dagegen dem System alle gewünschten Links hinzugefügt. Dabei besteht die Option, Links in Kategorien einzuordnen, eine eindeutige und zutreffende Beschreibung zu wählen, das Verhalten beim Klicken zu bestimmen und eine Beziehung zum jeweiligen Link zu vergeben.

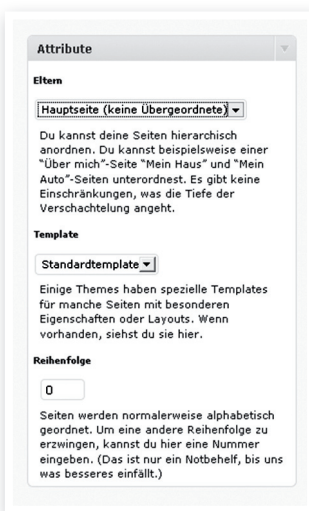
In dem Untermenüpunkt **LINK KATEGORIEN** können die **KATEGORIEN** der Links erstellt und verwaltet werden. Auch dieser Bereich ist ähnlich aufgebaut, wie die Kategorien für Artikel. Linkkategorien lassen sich dadurch differenzieren, dass sie nicht verschachtelt werden können.¹⁵⁰

Abbildung 4.13: LINKS HINZUFÜGEN

¹⁵⁰ Adler, 2009, S. 47

4.2.5 SEITEN

Im Menüpunkt SEITEN der Administrationsoberfläche können dem System statische SEITEN hinzugefügt werden. Im Gegensatz zu den ARTIKELN gehören SEITEN keiner KATEGORIE an und werden nicht chronologisch geordnet. Die Darstellung ist deshalb eigenständig und außerhalb der chronologischen Wiedergabe¹⁵¹. Das Einsatzgebiet für SEITEN sind vor allem die Vermittlung und Darstellung zeitunabhängiger und gleichbleibender Inhalte. Innerhalb von SEITEN können durch PLUGINS, Erweiterungen eingebunden werden, welche über die mitgelieferten Module steuerbar sind. Die Integration von SEITEN besteht seit der WordPress Version 1.5 und ist einer der wichtigsten Merkmale, um WordPress als WCMS einzusetzen. Durch Separation auf Unterseiten und Festlegung einer Seitenhierarchie können komplexe Inhalte übersichtlich in WordPress erstellt werden. Ein weiterer Vorteil und typisches WCMS-Merkmal ist die Verwendung eigener SEITEN-TEMPLATES.¹⁵²



Die Einbindung seitenspezifischer TEMPLATES erfolgt bei der Erstellung von SEITEN im Backend. In diesem Punkt unterscheiden sich auch die beiden Eingabemasken voneinander. Bei der Einrichtung von SEITEN können, im Gegensatz zu den ARTIKELN, über das zusätzliche Bedienfeld ATTRIBUTE, SEITENTEMPLATES gewählt werden. Innerhalb dieses Bedienfeldes werden auch die Eltern- und Kinderseiten sowie die Reihenfolge für die Erscheinung im Frontend bestimmt.¹⁵³ Besonders nützlich ist bei SEITENTEMPLATES, dass eigene PHP-Skripte und Funktionen, die über den herkömmlichen Umfang von SEITEN hinausgehen, eingesetzt werden können. Beispielsweise ist auf diese Art und Weise die Realisierung eines Gästebuches möglich.¹⁵⁴

Abbildung 4.14: Bedienfeld ATTRIBUTE im Menüpunkt SEITE ERSTELLEN

Unter dem Menüpunkt SEITE BEARBEITEN sieht man ähnlich wie bei ARTIKELN eine Gesamtübersicht der vorhandenen SEITEN. Von hier aus können die SEITEN bearbeitet, angesehen und gelöscht werden. Bestehende Seitenentwürfe werden mit dem Vermerk ENTWURF gekennzeichnet. Im Gegensatz zu den ARTIKELN findet man hier keine Kategorisierung. Unterseiten werden mit einem Anstrich dargestellt.

¹⁵¹ Vgl. Mullenweg, 2010 [www-dokument] - Pages

¹⁵² Vgl. Mullenweg, 2010 [www-dokument] - Pages

¹⁵³ Vgl. o.A., 2009 [www-dokument] - Page Attributes

¹⁵⁴ Vgl. Bültge, 2009, S. 64

4.2.6 KOMMENTARE

Ähnlich wie bei ARTIKELN und SEITEN erhält man im Bereich KOMMENTARE einen Überblick der vorhandenen Kommentare zu Beiträgen oder eventuellen Gästebucheinträgen. In dieser Übersicht können Kommentare als Spam markiert, gelöscht und bearbeitet werden. Auch das Verfassen einer Antwort ist in diesem Rahmen möglich.

Die Darstellung von Kommentaren im Frontend fällt in der Regel relativ unterschiedlich aus. Viele Weblogs ermöglichen das Verfassen von Kommentaren unterhalb der geschriebenen Artikel. Dabei wird der Öftere der neueste Kommentar als Oberster gelistet. Für das Verfassen von Kommentaren sind die Nutzer aufgefordert, Name und E-Mail Adresse anzugeben.¹⁵⁵

Um sich vor Kommentarspam zu schützen, bietet es sich zudem an, CAPTCHAs einzusetzen. Diese werden in vielen Fällen über simple Rechenaufgaben realisiert. Die Konfiguration von Kommentaren erfolgt über den Menüpunkt Einstellungen–Diskussionen im Backend.

4.2.7 DESIGN

Der Punkt DESIGN im Administrationsmenü umfasst das generelle Erscheinungsbild der Website im Frontend. Dabei kann zwischen den Unterpunkten WIDGETS, EDITOR, INSTALLATION und PERSONALISierter HEADER selektiert werden. Das Erscheinungsbild wird in DESIGNTHEMES festgelegt. THEMES sind gebündelte TEMPLATE-Dateien, die Layout und Inhalt voneinander trennen wobei alle design-spezifischen Elemente via HTML bestimmt und per CSS gestylt werden.¹⁵⁶

Inhaltsspezifische Elemente gelangen über PHP und den WordPress eigenen TEMPLATE TAGS sowie CONDITIONAL TAGS aus der Datenbank in das Frontend.¹⁵⁷ Ein THEME besteht aus mindestens einer Styledatei („style.css“) und einer Indexdatei („index.php“). Gerade in Bezug auf WordPress-basierte WCMS-Lösungen, sollten TEMPLATES erweitert werden, sodass der Überblick über den Code und dessen Funktionen erhalten bleibt. Der grundlegende Aufbau und die Funktionsweisen von TEMPLATE-Dateien und THEMES wird im Kapitel 5 explizit untersucht. Im Standardtheme KUBRICK 1.6 besteht das THEME aus folgenden TEMPLATES¹⁵⁸:

¹⁵⁵ Vgl. Bültge, 2009, S. 20

¹⁵⁶ Vgl. o.A., 2009 [www-dokument] - Was ist ein THEME?

¹⁵⁷ Vgl. Bültge, S. 2009, S 31

¹⁵⁸ Vgl. Heilemann, 2010, [www-dokument]

- 404.php
- archive.php
- archives.php
- comments.php
- comments-popup.php
- footer.php
- functions.php
- header.php
- image.php
- index.php
- links.php
- page.php
- search.php
- sidebar.php
- single.php
- style.css
- rtl.css
- de_DE.mo
- de_DE.po
- screenshot.png
- images Verzeichnis

Im Backend können unter dem Punkt **DESIGN /THEMES**, die verfügbaren **THEMES** aktiviert werden. In der Regel findet man hier drei verschiedene **THEMES** zur Auswahl. Das **STANDARDTHEME CLASSIC 1.5** von **DAVE SHEA** oder **DEFAULT 1.6 THEME** von **MICHAEL HEILEMANN** sowie das eigene beziehungsweise das erworbene **THEME**.

Im Unterpunkt **WIDGET** können dem **THEME** verschiedene **SIDEBAR WIDGETS** per ‚drag and drop‘ hinzugefügt werden. Allerdings unterstützen einige **THEMES** diese Funktion nicht immer automatisch, sodass die Einbindung von den gewünschten **WIDGETS** nicht zum Ziel im Frontend führt. Um die minimalste Funktionsweise von **WIDGETS** zu nutzen, muss in der Datei ‚functions.php‘ folgender Code eingebunden werden¹⁵⁹:

159 Vgl. Zelend, 2008, [www-dokument]

```
01  <?php
02      if (function_exists('register_sidebar') )
03          register_sidebar();
04  ?>
```

Listing 4.7: Registrierung der **SIDEBAR** in der Datei **,function.php'** des **THEMES**

```
01  <ul>
02  <?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar() ) : endif; ?>
03  </ul>
```

Listing 4.8: Implementierung der **SIDEBAR** in der Datei **,sidebar.php'** des **THEMES**

Im Unterpunkt **EDITOR** können unter der Voraussetzung, dass der Server Schreibrechte auf die Datei besitzt, die **THEMEDATEIEN** bearbeitet und gespeichert werden¹⁶⁰. Dieser **EDITOR** funktioniert allerdings ohne **SYNTAX HIGHLIGHTING** und wird nach dem Speichern direkt im Frontend umgesetzt. Aus diesem Grund sind vor allem Desktop-Anwendungen bei der Bearbeitung und Entwicklung von **THEMES** zu empfehlen, um beispielsweise PHP-Fehler zu vermeiden.

4.2.8 PLUGINS

Der Reiter **PLUGINS** im Administrationsbereich zählt mit zu den Wichtigsten, um **WORDPRESS** als **WCMS** nutzbar zu machen. Von hier aus werden Programmiererweiterungen eingebunden, die den Funktionsumfang erheblich steigern, ohne den Programmkern des Systems zu manipulieren¹⁶¹. Es wird damit gleichzeitig eine Weiche zwischen Aktualisierung und optimaler Geschwindigkeit gestellt. Würde **WORDPRESS** beispielsweise alle **WCMS-Features** im Kern beinhalten, wäre die Performance unter Umständen langsamer. Mit Hilfe von **PLUGINS** umgeht man einen Geschwindigkeitsverlust und somit ständiges Abfragen von Datenbank-Tabellen oder das Ausführen von Skripten.¹⁶²

¹⁶⁰ Vgl. Mullenweg, 2010 [www-dokument] - Changing File Permissions

¹⁶¹ Vgl. o.A., 2010 [www-dokument]- Plug-in Übersicht

¹⁶² Vgl. o.A., 2010 [www-dokument]- Plug-in Übersicht

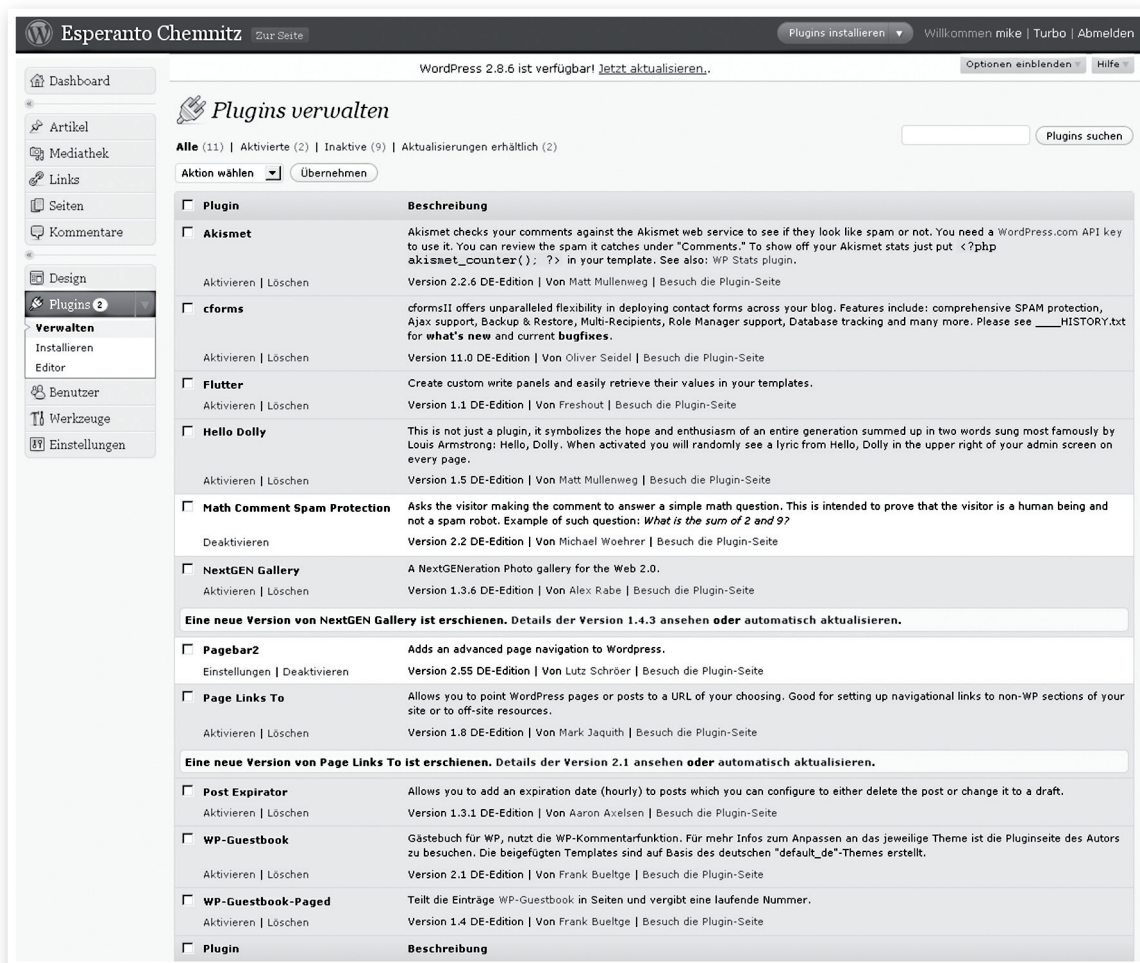


Abbildung 4.15: PLUGINVERWALTUNG

Im Bereich **PLUGINVERWALTUNG** erhält der Administrator eine Gesamtübersicht der vorhandenen und aktiven **PLUGINS**. Aus diesem Bereich lassen sich **PLUGINS** aktivieren, deaktivieren sowie löschen. Falls eine neue Version des bereits installierten **PLUGINS** verfügbar ist, erhält man einen entsprechenden Hinweis im Menü. Außerdem wird in einer Kurzbeschreibung der Funktionsumfang des **PLUGINS** erläutert.

Die Installation von **PLUGINS** erfolgt über das Hochladen der Dateien zum Webserver via FTP in das Verzeichnis `.wp-content/plugins/`. **WORDPRESS** bietet allerdings auch die Option, die Daten direkt aus dem Backend zu importieren. Hierzu müssen in der Administrationsoberfläche die Zugangsdaten zum Server hinterlegt werden, um den Prozess zu vollziehen.

Nachdem sich die **PLUGIN**-Daten auf dem Server befinden, muss die Erweiterung im Backend aktiviert und konfiguriert werden¹⁶³. Die Konfiguration kann dabei von **PLUGIN** zu **PLUGIN** unterschiedlich ablaufen. Größere **PLUGINS** legen im Backend beispielsweise einen eigenen Menüpunkt an. Kleine Erweiterungen nutzen dagegen die **EINSTELLUNG** von **WORDPRESS**.

PLUGINS können von den offiziellen **WORDPRESS** Betreiberseiten oder von Drittanbieter bezogen werden. Jedoch sollte man dabei bedenken, dass jedes **PLUGIN** von Drittanbieter nicht automatisch standardkonform programmiert wurde und Sicherheitslücken enthalten kann.

4.2.9 BENUTZER

Im Menüpunkt **BENUTZER** findet man die Benutzerverwaltung von **WORDPRESS**. Hier erhält man einen Gesamtüberblick der bestehenden Benutzer und ihren jeweiligen Rechten. Es können unterschiedlich viele **BENUTZER** im System angelegt und administriert werden. Diesen **BENUTZERN** werden Rollen vergeben, sodass es beispielsweise möglich ist, dass sich **REDAKTEURE** dem Publizieren von Inhalten widmen, **AUTOREN** und **MITARBEITER** redaktionelle Texte einpflegen und **ADMINISTRATOREN**, sich mit der Darstellung und dem Funktionsablauf des Systems befassen.¹⁶⁴

Der folgende Überblick stellt die Rechte und Fähigkeiten der jeweiligen Benutzer dar.¹⁶⁵

Fähigkeit	Administrator	Redakteur	Autor	Mitarbeiter	Leser
Themes installieren	X	-	-	-	-
Theme aktualisieren	X	-	-	-	-
Themes wechseln	X	-	-	-	-
Themes bearbeiten	X	-	-	-	-
Plugins installieren	X	-	-	-	-
Plugins aktivieren	X	-	-	-	-
Plugins bearbeiten	X	-	-	-	-
Plugins aktualisieren	X	-	-	-	-
Plugins löschen	X	-	-	-	-
Benutzer erstellen	X	-	-	-	-
Benutzer bearbeiten	X	-	-	-	-
Benutzer löschen	X	-	-	-	-

¹⁶³ Vgl. o.A., 2010 [www-dokument] - Plug-in Übersicht

¹⁶⁴ Vgl. Mullenweg, 2010 [www-dokument] - Users Your Profile SubPanel

¹⁶⁵ Vgl. Bültge, 2009, S. 81 ff.

Dateien bearbeiten	X	-	-	-	-
Importieren	X	-	-	-	-
Ungefiltertes Hochladen	X	-	-	-	-
Dashboard bearbeiten	X	-	-	-	-
Kommentare moderieren	X	X	-	-	-
Kategorien verwalten	X	X	-	-	-
LINKS verwalten	X	X	-	-	-
Ungefiltertes HTML	X	X	-	-	-
Bearbeiten von pub. Beiträgen	X	X	-	-	-
Andere Beiträge bearbeiten	X	X	-	-	-
Seiten bearbeiten	X	X	-	-	-
Bearbeiten von publizierten Seiten	X	X	-	-	-
Andere Seiten bearbeiten	X	X	-	-	-
Seiten veröffentlichen	X	X	-	-	-
Seiten löschen	X	X	-	-	-
Andere Seiten löschen	X	X	-	-	-
Löschen veröffentlichter Seiten	X	X	-	-	-
Löschen von anderen Artikel	X	X	-	-	-
Private Artikel löschen	X	X	-	-	-
Bearbeiten privater Artikel	X	X	-	-	-
Private Artikel lesen	X	X	-	-	-
Private Seiten löschen	X	X	-	-	-
Bearbeiten privater Seiten	X	X	-	-	-
Lesen privater Seiten	X	X	-	-	-
File hochladen	X	X	X	-	-
Artikel veröffentlichen	X	X	X	-	-
Löschen veröffentlichter Artikel	X	X	X	-	-
Artikel bearbeiten	X	X	X	X	-
Beiträge löschen	X	X	X	X	-
Lesen	X	X	X	X	X

Anhand der Tabelle wird deutlich, welche Rechteverwaltung in WordPress möglich ist. Es besteht des Weiteren die Option, dass sich User über ein Login-Formular registrieren können. Allerdings sollte dabei die Rolle des Benutzers und dessen Rechte genau betrachtet und vergeben werden, womit eine generelle Sicherung des Systems gewahrt bleibt. Eine detaillierte Vergabe von Rechten und die benutzerdefinierte Anpassung des Backends ist mit dem Plugin WP-ADMINIMIZE möglich.¹⁶⁶

¹⁶⁶ Siehe: <http://wordpress.org/extend/plugins/adminimize/>

4.2.9.1 Hinzufügen von AUTOREN und BENUTZERN

In diesem Bereich des Systems können BENUTZER über das Backend angelegt werden. Hierzu müssen lediglich die definierten Inputfelder ausgefüllt, ein Passwort und die Rolle des Benutzers vergeben werden. Nachdem der BENUTZER angelegt wurde, kann im Reiter DEIN PROFIL der genaue Status des Nutzers dokumentiert werden. Es wird beispielsweise festgelegt, in welcher Farbgebung das Backend dargestellt wird, sowie die Erfassung von persönlichen Daten und die Kontaktmöglichkeiten des Autors.

4.2.10 WERKZEUGE

Im Bereich WERKZEUGE kann die Verwendung von BOOKMARKLETS und die lokale Backend Beschleunigung mit Hilfe von GOOGLE GEARS installiert werden. Der Bereich bietet ebenso Schnittstellen, womit ARTIKEL oder KOMMENTARE aus anderen Blogsystemen importiert und exportiert werden können¹⁶⁷. Beim Import von anderen Blogsystemen, wie zum Beispiel LIVEJOURNAL, TEXTPATTERN oder ähnliche wird für das jeweilige System eine spezielle Importoption angeboten. Bei Import von anderen WORDPRESS-Installationen wird auf die Schnittstelle EXTENDED RSS zurückgegriffen. Diese wird mittels einer XML-Datei beim Export von WORDPRESS erzeugt. Die Exporteinstellung wird nach Benutzern begrenzt und beinhaltet ARTIKEL, SEITEN, KOMMENTARE, BENUTZERDEFINIERT FELD, KATEGORIEN sowie TAGS.¹⁶⁸

Seit der Version 2.7 besteht die Möglichkeit, WORDPRESS automatisch über das Backend zu aktualisieren, wobei die Zugangsdaten zum Webserver in der Administrationsoberfläche hinterlegt werden müssen¹⁶⁹. Wie vor jeder Aktualisierung sollte auch beim automatischen Upgrade ein Backup des Systems erstellt werden.¹⁷⁰

4.2.11 EINSTELLUNG

In diesem Teil des Systems befinden sich die globalen Konfigurationsmöglichkeiten von WORDPRESS. Der Einstellungsbereich wird in acht unterschiedliche Bereiche geteilt. Diese Segmente werden im Folgenden kurz beschrieben und deren Funktionsweise und Bedeutung für ein WCMS ergründet.

167 Vgl. Schmitz, 2009 [www-dokument] – WORDPRESS 2.7 wurde veröffentlicht

168 Vgl. Mullenweg, 2010 [www-dokument] – Administration Panels

169 Vgl. Schmitz, 2009 [www-dokument] – WORDPRESS 2.7 wurde veröffentlicht

170 Vgl. Mullenweg, 2010 [www-dokument] – Updating WORDPRESS

4.2.11.1 ALLGEMEIN

Im Reiter ALLGEMEIN werden die universellen Informationen des Systems hinterlegt. Dies können zum Beispiel der BLOGTITEL, SLOGAN, WORDPRESS ADRESS-URL, BLOG-URL, E-MAIL und REGISTRIERUNGSBEDINGUNGEN sein. Der TITEL und SLOGAN kann bei der Implementierung im THEME ausgegeben werden. Dies wird beim TITEL entweder über den TEMPLATE TAG ‚wp_title‘ oder über ‚bloginfo‘ realisiert.

```
01 <h1><?php bloginfo('name'); ?></h1>
```

Listing 4.9: TEMPLATE TAG ‚bloginfo‘

Die im Reiter ALLGEMEIN hinterlegte E-Mail Adresse dient hauptsächlich als administrative Zugriffs- und Benachrichtigungsquelle, falls beispielsweise KOMMENTARE ZU ARTIKELN veröffentlicht wurden oder wenn sich neue Nutzer im System registrieren¹⁷¹. Auch die Bestimmung über Zeit-zonen und Datumsausgaben wird im Bereich ALLGEMEIN mittels der PHP-Funktion date() realisiert.¹⁷²

4.2.11.2 SCHREIBEN und LESEN

In dem Unterpunkt SCHREIBEN der Systemeinstellungen wird festgelegt, welche Zeilengröße der EDITOR einnimmt, ob fehlerhafter HTML-Code im EDITOR korrigiert wird oder ob EMOTIONS in Smiley-Icons umgewandelt werden sollen. Des Weiteren ist an dieser Stelle einstellbar, ob ARTIKEL über das Bedienfeld FERNPUBLIZIEREN via ATOM PUBLISHING PROTOCOL, XML-RPC-Schnittstelle oder via E-Mail publiziert werden sollen.

Im Bereich LESEN kann die Darstellung von SEITEN und ARTIKEL in WORDPRESS bestimmt werden. Dabei kann sowohl die Startseite als Artikelseite genutzt werden, als auch eine bestimmte (wählbare) statische SEITE. In der Praxis ist es bei Weblogs üblich, dass die Startseite der Artikelseite entspricht. Dagegen ist bei WCMS realisierten Websites diese Festlegung ungewöhnlich. Vielmehr werden hier Begrüßungstexte oder Vorstellungen auf der Startseite transportiert. Diese Festlegung der Startseite besteht seit der Version 2.1 und beschreibt ein weiteres Indiz für die Verwendung von WORDPRESS als WCMS.¹⁷³

¹⁷¹ Vgl. o.A. 2010 [www-dokument] - Allgemeine Einstellungen

¹⁷² Vgl. o.A. 2010 [www-dokument] - date

¹⁷³ Vgl. Bültge, 2009, S. 87

4.2.11.3 DISKUSSIONEN, MEDIATHEK und PRIVATSPHÄRE

Die Hauptaufgabe des Punktes DISKUSSIONEN ist die Bestimmung der Kommentarfunktionalität von WORDPRESS. Es kann festgelegt werden, ob KOMMENTARE generell von Lesern, mit oder ohne Registrierung erstellt werden dürfen. Hieraus ergeben sich weitere Einschränkungsoptionen, wie zum Beispiel die Moderation und Freischaltung seitens des Administrators. Auch der Veröffentlichungszeitraum, sowie die Darstellung und Verteilung der KOMMENTARE auf mehreren Seiten wird diesem Bereich fixiert.

Die Einstellung im Bereich der MEDIATHEK beziehen sich hauptsächlich auf die Darstellungsgröße von Thumbnails. Es kann festgelegt werden, in welcher Größe im Frontend ausgegeben werden. Dabei skaliert WORDPRESS die Bilder immer proportional.

4.2.11.4 PERMALINKS

Innerhalb diesen Reiters können suchmaschinenfreundliche URLs erzeugt werden. Dabei hat man fünf unterschiedliche Optionen zur Auswahl. Die Permalink-Standardeinstellung hat den Vorteil, dass Links, ohne aktiviertes ‚mod_rewrite‘-Modul des Webserver dargestellt werden¹⁷⁴. Jedoch sollte bei der Existenz des ‚mod_rewrite‘ Modul diese Darstellung möglichst nicht verwendet werden, da sie für Suchmaschinen und für den Menschen nicht schlecht interpretierbar ist.

Das Apache-Modul ‚mod_rewrite‘ ermöglicht mit Hilfe von ‚regulären Ausdrücken‘ die Manipulation und Umschreibung der URLs¹⁷⁵. Somit können suchmaschinenfreundliche URLs erzeugt werden. Diese Option wird innerhalb der Backend-Einstellung als BENUTZERDEFINIERTER STRUKTUR betitelt. WORDPRESS bietet in dieser BENUTZERDEFINIERTEN STRUKTUR die Möglichkeit, unter der Verwendung von PERMALINK TAGS, eine eigenständige Linkstruktur zu erzeugen¹⁷⁶. Hierbei kann das JAHR, MONAT, TAG, STUNDE, MINUTE, SEKUNDE, BEITRAGSNAME, ID-NUMMER, AUTOR, KATEGORIE und TAG in die Struktur eingebaut werden. Da PERMALINKS die Performance des Systems gerade bei vielen SEITEN schwächen können, bietet es sich an, möglichst kurze Namen für die Links zu definieren.¹⁷⁷

¹⁷⁴ Vgl. Alby, 2008a, S. 257

¹⁷⁵ Vgl. Bültge, 2009, S. 89

¹⁷⁶ Vgl. Mullenweg, 2010 [www-dokument] – Using Permalinks

¹⁷⁷ Vgl. Bültge, 2009, S. 91

5. THEMES und TEMPLATES

Nachdem die wichtigsten Backendfunktionen und Eigenschaften für die Nutzung von WordPress als WCMS erläutert wurden, soll in diesem Kapitel die Architektur sowie die Wirkungsweise des Programmcodes analysiert werden.

Für die WCMS-Nutzung von WordPress spielen THEMES und dessen TEMPLATE-Dateien eine entscheidende Rolle. THEMES werden auf dem Webserver im Verzeichnis `.root/wp-content/themes/THEMENAME` abgelegt. Sie bestehen mindestens aus zwei Dateien, der `.style.css` und der `.index.php`. Sollten diese beiden Dateien nicht vorhanden sein, betrachtet WordPress das THEME als defekt und gibt eine Fehlermeldung aus.¹⁷⁸

In einem THEME werden sämtliche TEMPLATES, zusammenhängende PHP-Codeblöcke, und HTML-Tags miteinander verknüpft, die wiederum auf die Style-Datei zurückgreifen oder im Fall von TEMPLATES mit der Datenbank kommunizieren. In den meisten Fällen befindet sich ein weiteres Verzeichnis mit dem Titel `.images` im THEME-Ordner, welches allgemeine Layoutgrafiken beinhaltet¹⁷⁹. Zusätzlich sollte sich im THEME-Ordner ein `.screenshot.png` befinden, welcher im Menüpunkt DESIGN die Zuordnung verschiedener THEMES und Layouts ermöglicht. Zusammenfassend werden in einem THEME, Design und Programmlogik verknüpft und im Frontend dargestellt.

5.1 Aufbau der Datei `.index.php` des DEFAULT-THEMES

Um einen Einblick in das Zusammenspiel von Layout und die Funktionsweise des Programmcodes zu erhalten, soll im Kapitelabschnitt 5.1 die Architektur der `.index.php` des DEFAULT-THEMES analysiert werden. Darüber hinaus findet eine Erläuterung zu den Begriffen TEMPLATE TAG, CONDITIONAL TAG und LOOP statt.

```
01 <?php
02 /**
03  * @package WordPress
04  * @subpackage Default_Theme
05  */
06 get_header(); ?>
07 <div id="content" class="narrowcolumn" role="main">
```

¹⁷⁸ Vgl. Bülgte, 2009, S. 180

¹⁷⁹ Vgl. o.A., 2009 [www-dokument] - Was ist ein THEME?

```

08  <?php if (have_posts()) : ?>
09      <?php while (have_posts()) : the_post(); ?>
10          <div <?php post_class(); ?> id="post-<?php the_ID(); ?>">
11              <h2><a href="<?php the_permalink() ?>" rel="bookmark"
12                  title="<?php printf(__('Permanent Link to %s', ,kubrick'), the_title_
13                  attribute(echo=0')); ?>"><?php the_title(); ?></a></h2>
14              <small><?php the_time(__('F jS, Y', ,kubrick')) ?> <!-- by <?php
15                  the_author() ?> --></small>
16              <div class="entry">
17                  <?php the_content(__('Read the rest of this entry
18                  &raquo;', ,kubrick')); ?>
19              </div>
20              <p class="postmetadata"><?php the_tags(__('Tags:', ,kubrick')) . ,
21                  , , , , ,<br />'); ?> <?php printf(__('Posted in %s', ,kubrick'), get_the_category_
22                  list(, , )); ?> | <?php edit_post_link(__('Edit', ,kubrick'), , ', , | '); ?> <?php
23                  comments_popup_link(__('No Comments &#187;', ,kubrick'), __('1 Comment &#187;',
24                  ,kubrick'), __('% Comments &#187;', ,kubrick'), , ', __('Comments Closed', ,kubrick')
25                  ); ?></p>
26              </div>
27          <?php endwhile; ?>
28          <div class="navigation">
29              <div class="alignleft"><?php next_posts_link(__('&laquo; Older Entries',
30                  ,kubrick')) ?></div>
31              <div class="alignright"><?php previous_posts_link(__('Newer Entries
32                  &raquo;', ,kubrick')) ?></div>
33          </div>
34          <?php else : ?>
35              <h2 class="center"><?php _e('Not Found', ,kubrick'); ?></h2>
36              <p class="center"><?php _e('Sorry, but you are looking for something that
37                  isn't here.', ,kubrick'); ?></p>
38              <?php get_search_form(); ?>
39          <?php endif; ?>
40      </div>
41  <?php get_sidebar(); ?>
42  <?php get_footer(); ?>
43  ?>

```

Listing 5.1: Gesamtauszug der Datei ‚index.php‘ des DEFAULT-THEMES¹⁸⁰

¹⁸⁰ Siehe: <http://themes.wordpress-deutschland.org/kubrick-de-edition/>

Die Datei ‚index.php‘ des Standard-THEMES enthält in den ersten zwei Zeilen einen Kommentar, indem definiert wird, wofür die Datei Verwendung findet. Außerdem beinhaltet diese Angaben über das genutzte THEME. Nach dem Kommentar folgt die erste ‚Include-Anweisung‘, welche die Datei ‚header.php‘ in das THEME einbindet. Damit die Datei integriert werden kann, muss sie allerdings im Verzeichnis existieren, anderenfalls wird eine PHP-Fehlermeldung ausgegeben.

WORDPRESS definiert im Kern einige dieser ‚Include-Tags‘, welche ohne spezielle Angaben zum jeweiligen Pfad eingebunden werden können. Es handelt sich dabei um die TAGS ‚<?php get_header(); ?>‘ für den Kopfbereich, ‚<?php get_footer(); ?>‘ für den Fußbereich, ‚<?php get_sidebar(); ?>‘ für die Sidebar, ‚<?php get_search_form(); ?>‘ für die Suchabfrage und ‚<?php comments_template(); ?>‘ für die Kommentare. Neben diesen bereits definierten Einbindungen können weitere Dateien mit der regulären PHP ‚Include-Anweisung‘ ‚<?php include(TEMPLATEPATH . ‚/sidebar_secondary.php‘); ?>‘ implementiert werden¹⁸¹. Die Einbindung der verschiedenen ausgelagerten Dateien, erfolgt aufgrund einer besseren Übersichtlichkeit in der Datei ‚index.php‘.

Nachdem die Header-Datei in das THEME eingebunden ist, folgt ein HTML-Tag mit einer eindeutigen ‚ID‘ und einer ‚Klasse‘, die über die Datei ‚style.css‘ und deren Formatdefinition gestylt werden kann. Darauf folgend findet man die erste dynamische Abfrage-Schleife von WORDPRESS, der sogenannte LOOP, vor. Der LOOP beginnt mit¹⁸²:

```
01  <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
```

Listing 5.2: Beginn des Loops

```
01  <?php endwhile; else: ?>
02  <p><?php _e(„Sorry, no posts matched your criteria.“); ?></p>
03  <?php endif; ?>
```

Listing 5.3: Ende des Loops

¹⁸¹ Vgl. Mullenweg, 2010 [www-dokument] – Include Tags

¹⁸² Vgl. Mullenweg, 2010 [www-dokument] – The Loop

```

01  <?php if (have_posts()) : ?>
02      <?php while (have_posts()) : the_post(); ?>
03          <div <?php post_class(); ?> id="post-<?php the_ID(); ?>">
04              <h2><a href="<?php the_permalink() ?>" rel="bookmark"
05                  title="<?php printf(__('Permanent Link to %s', ,kubrick'), the_title_
06                  attribute(,echo=0`)); ?>"><?php the_title(); ?></a></h2>
07                  <small><?php the_time(__('F jS, Y', ,kubrick')) ?> <!-- by
08                  <?php the_author() ?> --></small>
09                  <div class="entry">
10                      <?php the_content(__('Read the rest of this entry
11                      &raquo;', ,kubrick')); ?>
12                  </div>
13                  <p class="postmetadata"><?php the_tags(__('Tags:',
14                  ,kubrick') . , , , , , , , ,<br />'); ?> <?php printf(__('Posted in %s', ,kubrick'),
15                  get_the_category_list(, ,)); ?> | <?php edit_post_link(__('Edit', ,kubrick'), ,', ,
16                  | ,); ?> <?php comments_popup_link(__('No Comments &#187;', ,kubrick'), __('1 Comment
17                  &#187;', ,kubrick'), __(' % Comments &#187;', ,kubrick'), ,', __('Comments Closed',
18                  ,kubrick') ); ?></p>
19              </div>
20          <?php endwhile; ?>
21      <div class="navigation">
22          <div class="alignleft"><?php next_posts_link(__('&laquo; Older
23          Entries', ,kubrick')) ?></div>
24          <div class="alignright"><?php previous_posts_link(__('Newer
25          Entries &raquo;', ,kubrick')) ?></div>
26      </div>
27  <?php else : ?>
28      <h2 class="center"><?php _e('Not Found', ,kubrick'); ?></h2>
29      <p class="center"><?php _e('Sorry, but you are looking for something that
30      isn't here.', ,kubrick'); ?></p>
31      <?php get_search_form(); ?>
32  <?php endif; ?>

```

Listing 5.4: Die Gesamtdarstellung des `LOOPS` innerhalb der `index.php` des `DEFAULT-THEMES`

5.1.1 Der Loop

Der Loop ist für die Darstellung und Ausgabe von ARTIKELN und SEITEN zuständig. Er beginnt mit einer ‚while-Schleife‘, bei denen alle angegebenen Kriterien aus der Datenbank abgefragt und ausgegeben werden. Dieser Prozess vollzieht sich so lange, bis keine Ergebnisse der Abfrage mehr vorliegen. Es kann jedoch auch vorkommen, dass die Abfrage keine Ergebnisse der Datenbank ausliefert.¹⁸³

Zu Beginn des Loops findet eine Überprüfung über die Existenz der Beiträge statt. Hierfür wird ein ‚if-Konstrukt‘ mit einer ‚have_posts()‘ Funktion abgefragt. Falls relevante Inhalte in der Datenbank gespeichert wurden, so wird die ‚while-Schleife‘ gestartet bis zu dem Punkt, an dem keine Beiträge mehr zur Verfügung stehen und die Schleife mit ‚<?php endwhile; ?>‘ endet.¹⁸⁴

Der Aufruf der Anweisung ‚the_post()‘ wird gleichzeitig in dieser Schleife abgefragt. Ohne diese Anweisung funktionieren die meisten TEMPLATE TAGS im Loop nicht. Durch ‚the_post()‘ wird zwar kein Inhalt generiert, jedoch beziehen sich viele dieser TEMPLATE TAGS auf diese Funktion. Die Ausgabe von Inhalten erfolgt erst mit dem TEMPLATE TAG ‚the_content()‘.¹⁸⁵

Falls keine Inhalte mit Hilfe der ‚while-Schleife‘ ausgegeben werden können, kommt die ‚else-Anweisung‘ des ‚if-Konstrukts‘ zur Ausführung. Im beschriebenen Fall wird die Ausgabe ‚<?php _e(„Not Found“, „kubrick“); ?>‘ eingesetzt. Mit der Funktion ‚_e()‘ oder auch ‚__e‘ wird eine Lokalisierung und Aufruf der Sprachdatei zur Übersetzung angefordert. Falls diese vorhanden ist, werden sämtliche Inhalte der Funktion übersetzt.¹⁸⁶

```
01 <div <?php post_class(); ?> id="post-<?php the_ID(); ?>">
02     <h2><a href="<?php the_permalink() ?>" rel="bookmark" title="<?php
printf(__ („Permanent Link to %s“, „kubrick“), the_title_attribute(„echo=0“));
?>"><?php the_title(); ?></a></h2>
03     // weiterer Inhalt, der hier nicht aufgeführt wird[...]
04 </div>
```

Listing 5.5: Darstellung eines ARTIKELS innerhalb des Loops

183 Vgl. Mullenweg, 2010 [www-dokument] – The Loop

184 Vgl. Alby, 2008, S.172

185 Vgl. Bültge, S.2009, S.191

186 Vgl. Simovic, 2007 [www-dokument]

In jedem Beitrag wird innerhalb des ‚div-Tags‘ eine ‚Klasse‘ und eine ‚ID‘ ausgegeben, welche über die ‚style.css‘ angesprochen und gestaltet werden kann. Durch den TEMPLATE TAG ‚the_permalink()‘, welcher innerhalb eines ‚H2‘ und ‚a-Tags‘ aufgerufen wird, verlinkt WordPress zum einzelnen Beitragsinhalt. Dieser Inhalt wird mit dem jeweiligen LOOP des TEMPLATES ausgegeben.¹⁸⁷

```
01 <div id="post-180" class="post-180 post hentry category-dates">
02   <h2><a title="Permalink zu Donnerstag, 29.10.2009, 20.30 Uhr" rel="bookmark"
href="http://exemple.com">Ueberschrift</a></h2>
03   <small>15. Oktober 2009 <!-- by mike --></small>
04   div class="entry">
05     <p>Mojito Night</p>
06   </div>
07 </div>
```

Listing 5.6: Fontend-Auflistung und Darstellung der einzelnen Beiträge

Der LOOP lässt sich durch bestimmte Anpassungen einschränken bzw. erweitern, sodass beispielsweise nicht alle Beiträge der bestehenden Kategorien ausgegeben werden. Auch die Anzahl und Auflistung der Beiträge, sowie die Sortierung kann im LOOP ermittelt werden. Einige diese Erweiterungen werden im Praxisteil Kapitel 6 erläutert.

5.1.2 TEMPLATE TAGS

TEMPLATE TAGS sind wichtige PHP-Funktionen, die den Darstellungsumfang von WordPress erweitern und die Nutzbarkeit als WCMS ermöglichen. TEMPLATE TAGS werden als Code-Fragmente in TEMPLATES eingebunden und rufen spezifische, dynamische Funktionen aus dem CORE von WordPress auf¹⁸⁸. Dabei lassen sich TEMPLATE TAGS individuell anpassen und mit bestimmten Parametern ausstatten. Diese TEMPLATE TAG-Parameter sind Variablen, welche bei der Ausführung in Funktionen übergeben werden.¹⁸⁹

¹⁸⁷ Vgl. Bültge, S.2009, S.190

¹⁸⁸ Vgl. Bültge, S.2009, S.197

¹⁸⁹ Vgl. Mullenweg, 2010 [www-dokument] – TEMPLATE TAGS

Man unterscheidet grundsätzlich drei verschiedene Arten von TEMPLATE TAGS¹⁹⁰:

1. TAGS ohne Parameter
2. TAGS mit ‚function-style‘ Parameter
3. TAGS mit ‚query-string-style‘ Parameter

5.1.2.1 TEMPLATE TAGS ohne Parameter

Einige TEMPLATE TAG-Funktionen benötigen keine weiteren Darstellungsmöglichkeiten und somit verwenden diese auch keine Parameter. Als Beispiel sind die TEMPLATE TAGS ‚the_meta()‘ oder ‚the_author_firstname()‘ anzuführen. Im Fall ‚the_meta‘ werden alle Inhalte des ‚Schlüssel/Werte‘-Paares aus den BENUTZERDEFINIERTEN FELDERN ausgegeben (siehe Kapitel 4.2.2.1). Die Implementierung entspricht der PHP-Syntax und wird deshalb zwischen den spitzen Klammern und den Fragezeichen eingebunden.¹⁹¹

5.1.2.2 TEMPLATE TAGS mit ‚function-style‘ Parameter

Eine Vielzahl von TEMPLATE TAGS die Parameter akzeptieren, verwenden die standardisierte PHP-Syntax¹⁹². Aus diesem Grund nutzt man auch die Bezeichnung ‚function-style Parameter‘. Es werden die Werte oder Parameterliste der Funktion übergeben, die in runden Klammern stehen. Dabei kann es auch vorkommen, dass Parameter durch Kommas getreten werden oder Konstanten beziehungsweise Variablen enthalten. Ein Beispiel für ein solches TEMPLATE TAG liefert ‚<?php bloginfo(.name‘); ?>‘. Dieser TAG legt mit dem sogenannten ‚show-Parameter‘ fest, welcher Inhalt ausgegeben wird. Es besteht aber auch die Möglichkeit, dem TEMPLATE TAG weitere Parameter wie ‚description‘, ‚admin_email‘ und ‚rss_url‘ mitzugeben.¹⁹³

Alle ‚show-Parameter‘ werden als String in Hochkommas übergeben. Bei INTEGERS spielen die Hochkommas keine Rolle – der Wert kann innerhalb oder außerhalb der Hochkommas stehen. Der BOOLEAN Parameter, kann mit TRUE/FALSE, oder mit entsprechenden numerischen Werten übermittelt werden.¹⁹⁴

Eine wichtige Rolle spielt die Reihenfolge und Auflistung der Parameter. Werden beispielsweise mehrere Parameter übergeben, so lassen diese sich durch Kommas trennen. Leere Parameter dagegen werden durch Hochkommas übermittelt.

¹⁹⁰ Vgl. Mullenweg, 2010 [www-dokument] – How to Pass Tag Parameters

¹⁹¹ Siehe: <http://php.net/manual/en/language.basic-syntax.php>

¹⁹² Vgl. o.A. 2010 [www-dokument] – Funktionsparameter

¹⁹³ Vgl. Mullenweg, 2010 [www-dokument] – Function Reference/bloginfo

¹⁹⁴ Vgl. Mullenweg, 2010 [www-dokument] – How to Pass Tag Parameters / Types of parameters

5.1.2.3 TAGS mit ‚query-string-style‘ Parameter

Die letzte Art der TEMPLATE TAGS nutzt eine Abfragezeichenkette, dessen Ähnlichkeit der Syntax einer PHP ‚GET-Methode‘ entspricht. Alle Funktionen, die einen großen Parameterumfang besitzen, werden mit einem ‚wrapper‘ (Schutzhülle) ausgestattet, wodurch verhindert wird, dass unnötig viele Parameter aufgelistet werden müssen. Als Beispiel dient der TEMPLATE TAG ‚wp_list_cats()‘, welcher als wrapper für ‚list_cats()‘ verwendet wird. Dieser Tag enthält 18 Parameter.¹⁹⁵

‚Query-string-style Parameter‘ sind vor allem nützlich wenn nur bestimmte Werte der Parameter geändert werden sollen. Parameter, die ihrer Standardwerte beibehalten, müssen nicht mit aufgeführt werden. Viele dieser TAG-Typen beginnen meist mit ‚wp_‘. Außerdem muss die Reihenfolge der Parameter innerhalb der TAGS nicht beachtet werden. Die Trennung der einzelnen Parameter erfolgt durch ein kaufmännisches UND (&).

```
01 <?php wp_list_authors(,show_fullname=1&feed=rss&optioncount=1'); ?>
```

Listing 5.7: Beispiel für die Auflistung mit ‚query-string-style‘ Parameter¹⁹⁶

5.1.3 CONDITIONAL TAGS

CONDITIONAL TAGS werden in WORDPRESS eingesetzt, um Bedingungen innerhalb eines TEMPLATES und dessen THEMES abzufragen. Mit diesen TAGS lassen sich lange oder komplexe PHP-Abfragen kompensieren und vereinfachen. So kann beispielsweise mit dem CONDITIONAL TAG ‚is_home()‘ abgefragt werden, ob es sich um die Homepage der Website handelt. Sollen zum Beispiel auf der Startseite andere Inhalte ausgegeben werden, so ist mit Hilfe dieses CONDITIONAL TAGS und einer ‚If-Abfrage‘ eine Fallunterscheidung realisierbar. Alle definierten CONDITIONAL TAGS geben entweder das Ergebnis TRUE oder FALSE zurück, wobei einige der TAGS auch Parameter enthalten können.¹⁹⁷

¹⁹⁵ Vgl. Mullenweg, 2010 [www-dokument] – Function Reference/ list cats

¹⁹⁶ Vgl. Mullenweg, 2010 [www-dokument] – How to Pass Tag Parameters

¹⁹⁷ Vgl. Mullenweg, 2010 [www-dokument] – CONDITIONAL TAGS

```
01  <?php if (is_home()){
02      echo „Die Startseite“;
03  }
04      else{
05      echo „Keine Startseite“;
06  };
07  ?>
```

Listing 5.8: Fallunterscheidung mit Hilfe von **CONDITIONAL TAGS**

Gerade für die Gliederung und Strukturierung einer **SIDEBAR** können diese **TAGS** im **WCMS**-Umfeld sehr hilfreich sein. Es wird verhältnismäßig wenig Quellcode erzeugt, der jedoch globale Auswirkung auf die einzelnen **TEMPLATES** hat. Ein Beispiel verdeutlicht die Datei ‚**sidebar.php**‘ des **DEFAULT-THEMES**. Ab der Zeile 01 wird fast ausschließlich mit **CONDITIONAL TAGS** gearbeitet¹⁹⁸:

```
01  <?php if ( is_404() || is_category() || is_day() || is_month() || is_year() ||
is_search() || is_paged() ) { ?>
02      <li>
03          <?php /* If this is a 404 page */ if (is_404()) { ?>
04          <?php /* If this is a category archive */ } elseif (is_
category()) { ?>
05              <p><?php printf(__ („You are currently browsing the archives for
the %s category.\", ,kubrick\"), single_cat_title(„\", false)); ?></p>
06              <?php /* If this is a yearly archive */ } elseif (is_day()) { ?>
07              <p><?php printf(__ („You are currently browsing the <a
href=\"%1$s/\">%2$s</a> blog archives for the day %3$s.\", ,kubrick\"), get_
bloginfo(„url\"), get_bloginfo(„name\"), get_the_time(__ („l, F jS, Y\", ,kubrick\"));
?></p>
08              <?php /* If this is a monthly archive */ } elseif (is_month()) {
?>
```

¹⁹⁸ Siehe: Ausschnitt Quellcode der Datei ‚**sidebar.php**‘ des **DEFAULT-THEMES**

```

09          <p><?php printf(__,(You are currently browsing the <a
href="%1$s/">%2$s</a> blog archives for %3$s.', ,kubrick'), get_bloginfo(,url'), get_
bloginfo(,name'), get_the_time(__,(F, Y', ,kubrick'))); ?></p>
10
11          <?php /* If this is a yearly archive */ } elseif (is_year()) { ?>
12          <p><?php printf(__,(You are currently browsing the <a
href="%1$s/">%2$s</a> blog archives for the year %3$s.', ,kubrick'), get_
bloginfo(,url'), get_bloginfo(,name'), get_the_time(,Y')); ?></p>
13          <?php /* If this is a monthly archive */ } elseif (is_search()) {
?>
14          <p><?php printf(__,(You have searched the <a href="%1$s/">%2$s</
a> blog archives for <strong>&#8216;%3$s&#8217;</strong>. If you are unable to find
anything in these search results, you can try one of these links.', ,kubrick'),
get_bloginfo(,url'), get_bloginfo(,name'), wp_specialchars(get_search_query(), true));
?></p>
15          <?php /* If this is a monthly archive */ } elseif (isset($_
GET[,paged']) && !empty($_GET[,paged'])) { ?>
16          <p><?php printf(__,(You are currently browsing the <a
href="%1$s/">%2$s</a> blog archives.', ,kubrick'), get_bloginfo(,url'), get_
bloginfo(,name')); ?></p>
17          <?php } ?>

```

Listing 5.9: Ausschnitt aus der Datei ‚sidebar.php‘ des DEFAULT-THEMES

5.1.3.1 Verwendung von CONDITIONAL TAGS in der Datei ‚sidebar.php‘

Das obige Beispiel gibt in der ersten Zeile eine Bedingung unter Verwendung zahlreicher CONDITIONAL TAGS wider. Hierfür wird abgefragt, ob es sich um die Inhalte der TAGS ‚is_404()‘, ‚is_category()‘, ‚is_day()‘, ‚is_month()‘, ‚is_year()‘, ‚is_search()‘, ‚is_paged()‘ handelt. Die einzelnen TAGS werden dabei mit einem logischen ODER als Operator verknüpft. Wird bei der Bedingung der Wahrheitswert TRUE zurückgegeben, folgen weitere Bedingungen mit analogen CONDITIONAL TAGS. Die wichtigsten TAGS für die Verwendung von WORDPRESS als WCMS werden im Anhang dokumentiert.¹⁹⁹

¹⁹⁹ Vgl. Mullenweg, 2010 [www-dokument] – Conditional Tags

5.2 Aufbau der Datei ‚header.php‘ des DEFAULT-THEMES

Im Abschnitt 5.1 des Kapitels wurde der Aufbau der Datei ‚index.php‘ analysiert. Dabei wurde im oberen Teil des Dokumentes, der Kopfbereich der Seite, mit der PHP ‚Include-Anweisung‘ ‚<?php get_header(); ?>‘ inkl. Inhalt der Datei ‚header.php‘ eingebunden. Der Kopfbereich oder auch HEADER genannt, beinhaltet neben den HTML <head> Tags auch den Kopfbereich des Layouts.²⁰⁰

```

01  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" „http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
02  <html xmlns=»http://www.w3.org/1999/xhtml« <?php language_attributes(); ?>>
03  <head profile=»http://gmpg.org/xfn/11«>
04  <meta http-equiv=»Content-Type« content=»<?php bloginfo('html_type'); ?>;
charset=<?php bloginfo('charset'); ?>» />
05  <title><?php bloginfo('name');
06  if ( is_single() ) {
07      _e('&raquo; Blog Archive', 'kubrick');
08  }
09  wp_title(); ?></title>
10  <link rel=»stylesheet« href=»<?php bloginfo('stylesheet_url'); ?>» type=»text/css«
media=»screen« />
11  <link rel=»alternate« type=»application/rss+xml« title=»<?php printf(__('%s RSS
Feed', 'kubrick'), get_bloginfo('name')); ?>» href=»<?php bloginfo('rss2_url'); ?>» />
12  <link rel=»pingback« href=»<?php bloginfo('pingback_url'); ?>» />
13  <style type=»text/css« media=»screen«>
14
15  <?php
16  // Checks to see whether it needs a sidebar or not
17  if (!empty($withcomments) && !is_single() ) {
18      ?>
19      #page { background: url(«<?php bloginfo('stylesheet_directory'); ?>/images/
kubrickbg-<?php bloginfo('text_direction'); ?>.jpg») repeat-y top; border: none; }
20  <?php } else { // No sidebar ?>
21      #page { background: url(«<?php bloginfo('stylesheet_directory'); ?>/images/
kubrickbgwide.jpg») repeat-y top; border: none; }
22  <?php } ?>
23  </style>

```

200 Vgl. Alby, 2008a, S.171

```
24 <?php wp_head(); ?>
25 </head>
26 <body>
27 <div id=»page«>
28 <div id=»header«>
29 <div id=»headerimg«>
30 <h1><a href=»<?php echo get_option('home'); ?>/«><?php bloginfo('name');
?></a></h1>
31 <div class=»description«><?php bloginfo('description'); ?></div>
32 </div>
33 </div>
34 <hr />
```

Listing 5.10: Ausschnitt aus der Datei ‚header.php‘ des DEFAULT-THEMES

Im oberen Drittel der Datei ‚header.php‘ wird der XHTML-Dokumententyp mit der Sprachvariante Transitional//EN festgelegt²⁰¹. Nach dieser Bestimmung, folgt der Aufruf der Funktion ‚<?php language_attributes(); ?>‘, welche die Werte `dir=„ltr“` und `lang=„de-DE“` ausgibt. Das Attribut ‚dir‘ bezieht den Wert ‚ltr‘ aus der Backendeinstellung und definiert die Leserichtung (left to right). Aus der Datei ‚wp-config.php‘ wird mit Hilfe des Attributs ‚lang‘, die festgelegte Spracheneinstellung ausgelesen.²⁰²

Nach dieser Festlegung folgt der HTML-Tag <head>. Innerhalb dieses TAGS werden die TAGS <meta>, <title>, <link> und <style> bestimmt. Diesen TAGS werden TEMPLATE- oder CONDITINAL TAGS mitgegeben. Vor allem der TEMPLATE TAG ‚bloggerinfo()‘ kommt hier in unterschiedlichster Form zum Einsatz. Dieser wird beispielsweise für die Darstellung der Zeichencodierung (‚charset‘), dem Content-Type (‚html_type‘), dem Pfad zur primären CSS-Datei (‚stylesheet_url‘) oder dem Aufruf der URL für PINGBACKS genutzt²⁰³. In früheren DEFAULT-THEMES wurden mit Hilfe von ‚bloggerinfo()‘ die WORDPRESS-Version im Quellcode signifiziert. Da jedoch die Ausgabe der Version ein Sicherheitsrisiko darstellen könnte, sollte die genutzte Version nicht mehr angegeben werden²⁰⁴. Nachdem der </style> TAG geschlossen wurde, folgt der erste WORDPRESS-HOOK mit ‚<?php wp_head(); ?>‘.

201 Siehe: <http://www.w3.org/TR/xhtml1/#dtds>

202 Vgl. Bülge, 2007, [www-dokument] - WP - Tutorial, wir schreiben ein WORDPRESS Theme - Teil 3

203 Vgl. Mullenweg, 2010 [www-dokument] - Function Reference/bloggerinfo

204 Vgl. Alby, 2008a, S.174

5.3 Hooks

HOOKS werden in WORDPRESS verwendet, um PLUGINS oder spezielle Funktionen in das System zu implementieren. Somit ist es möglich, Erweiterungsskripte, die für die Nutzung von WORDPRESS als WCMS von großer Bedeutung sind, einzubinden. Man differenziert dabei zwischen FILTER HOOKS und ACTION HOOKS. FILTER HOOKS (`.add_filter()`) kommen hauptsächlich zum Einsatz, wenn bestimmte Daten abgewandelt werden, bevor diese an den Client oder an die Datenbank gesendet werden. ACTION HOOKS (`.add_action()`) finden Verwendung, wenn bestimmte Aktionen eintreten, während der Ausführung von WORDPRESS²⁰⁵. Sie ermöglichen das Hinzufügen oder Löschen von Funktionen, die auf den CORE von WORDPRESS zurückgreifen. Der Einsatz von HOOKS soll vor allem Erleichterung bei der Übergabe von Daten bewirken und für eine sichere Einbindung von WCMS-Funktionen, durch PLUGINS, sorgen.²⁰⁶

5.3.1 Action Hooks

ACTION HOOKS bestimmen die Erstellung neuer Funktionen und dienen als Schnittstelle zwischen System und PLUGIN. Hierbei besteht jedoch nicht die Möglichkeit, Veränderungen oder Manipulationen der Daten zu generieren²⁰⁷. Um die Funktionsweise zu verdeutlichen, soll das folgende Beispiel einen Überblick des mitgelieferten PLUGINS HELLO DOLLY bilden.²⁰⁸

```
01  <?php
02  /**
03   * @package Hello_Dolly
04   * @author Matt Mullenweg
05   * @version 1.5
06   */
07  /*
08   PLUGIN Name: Hello Dolly
09   PLUGIN URI: http://wordpress.org/#
10   Description: [...]
11   Author: Matt Mullenweg
12   Version: 1.5
13   Author URI: http://ma.tt/
14   */
```

²⁰⁵ Vgl. Frütel, 2009, S.173

²⁰⁶ Vgl. Bültge, S.2009, S.241

²⁰⁷ Vgl. Frütel, 2009, S. 174

²⁰⁸ Siehe: <http://wordpress.org/extend/plugins/hello-dolly/>


```
15 function hello_dolly_get_lyric() {
16     /** These are the lyrics to Hello Dolly */
17     $lyrics = „Hello, Dolly
18     Well, hello, Dolly
19     It's so nice to have you back where you belong
20     You're lookin' swell, Dolly
21     I can tell, Dolly
22     [...] „;
23     // Here we split it into lines
24     $lyrics = explode(„\n“, $lyrics);
25
26     // And then randomly choose a line
27     return wptexturize( $lyrics[ mt_rand(0, count($lyrics) - 1) ] );
28 }
29 // This just echoes the chosen line, we'll position it later
30 function hello_dolly() {
31     $chosen = hello_dolly_get_lyric();
32     echo „<p id='dolly'>$chosen</p>“;
33 }
34 // Now we set that function up to execute when the admin_footer action is called
35 add_action(,admin_footer`, ,hello_dolly`);
36
37 // We need some CSS to position the paragraph
38 function dolly_css() {
39     echo „
40     <style type='text/css'>
41     #dolly {
42         position: absolute;
43         top: 4.5em;
44         margin: 0;
45         padding: 0;
46         right: 215px;
47         font-size: 11px;
48     }
49     </style>
50     „;
51 }
52 add_action(,admin_head`, ,dolly_css`);
53 ?>
```

Listing 5.11: PLUGIN HELLO DOLLY von MATT MULLENWEG

Im letzten Drittel des PLUGINS von MATT MULLENWEG werden die ACTION HOOKS definiert.

```
01  add_action('admin_footer', ,hello_dolly');  
02  [...]  
03  add_action('admin_head', ,dolly_css');
```

Listing 5.12: Ausschnitt aus dem PLUGIN HELLO DOLLY

Mit diesen beiden ACTION HOOKS werden die Funktionen ,hello_dolly()' und ,dolly_css()' in den Hooks der Administrationsoberfläche ,admin_head' und ,admin_footer' implementiert. Nach jedem Aufruf des Headers im Backend erfolgt die Einbindung der Funktion ,dolly_css()' mit der jeweiligen Styleanweisung. Im Footer wird die eigentliche inhaltsbasierende Funktion ,hello_dolly' verankert²⁰⁹. Wenn das PLUGIN beispielsweise nicht im Backend eingebunden werden soll, so kann es auch mit Hilfe der Hooks im Header (siehe Abschnitt 5.2) bzw. im Footer des Frontends eingefügt werden. Dazu müssten die ACTION HOOKS des PLUGINS wie folgt angepasst werden:

```
01  add_action('wp_footer', ,hello_dolly');  
02  [...]  
03  add_action('wp_head', ,dolly_css');
```

Listing 5.13: Individuelle Anpassung des PLUGINS HELLO DOLLY

Für die ACTION HOOKS gibt es zusätzliche, optionale Parameter wie beispielsweise die Vergabe einer Priorität. Diese entscheidet in welcher Reihenfolge die Funktionen ausgegeben werden sollen, falls mehrerer Positionen für den gleichen Hook definiert wurden.²¹⁰

²⁰⁹ Vgl. Frütel, 2009, S.174

²¹⁰ Vgl. Frütel, 2009, S.175 ff.

5.3.2 FILTER HOOKS

FILTER HOOKS ermöglichen die Beeinflussung und Manipulation von Daten, die an den Browser ausgegeben oder in die Datenbank geschrieben werden. WORDPRESS stellt für die Verwendung des Filters die Funktion ‚add_filter‘ bereit. So können beispielsweise mit der Funktion ‚count_words‘ und der Registrierung dieser Funktion im Filter, die Wörter eines Beitrages gezählt und am Ende des Artikels eingefügt werden. Dabei ist der erste Parameter der zu filternde Inhalt.²¹¹

```
01  <?php
02  function count_words ($content) {
03  $count = str_word_count($content);
04  $content .= "Anzahl der Woerter im Beitrag: $count";
05  return ($content);
06  }
07  add_filter('the_content', 'count_words');
08  ?>
```

Listing 5.14: Einsatz von FILTER HOOKS

211 Vgl. Frütel, 2009, S.178 f.

6. Praktischer Einsatz von WordPress als WCMS

Im folgenden Kapitel soll, nach den vorangegangenen theoretischen Analysen, der praktische Einsatz von WordPress als WCMS geprüft werden. Hierbei wird die Umsetzung einer mittelgroßen Website untersucht, die neben der Verwaltung und Pflege sämtlicher Inhalte, eine Bildergalerie, Kontaktformular, Gästebuch und Terminkalender mit Kommentarfunktion enthalten soll. Des Weiteren soll das in WordPress integrierte Framework JQUERY für Navigationselemente genutzt werden.

Eine realistische Umsetzung solcher Anforderungen bot sich mit der Website www.esperanto-network.de an. Hierbei wird im Folgenden nachvollzogen, wie sich WordPress als WCMS eignet und welche zusätzlichen PLUGINS implementiert werden müssen.

6.1 Anpassung von TEMPLATES

Schwerpunkt der Website ist die Darstellung aktueller Termine und Programmeinträge des Betreibers. Alle Veranstaltungen sollen nach Beendigung aus dem Frontend ausgeschlossen, jedoch im Backend innerhalb der ENTWÜRFE bestehen bleiben. Da in WordPress kein eigenes Kalendermodul im CORE des Systems integriert ist und vorhandene Kalender-PLUGINS²¹² nicht den gewünschten Funktionen entsprechen, wurde die Blogfunktionalität von WordPress angepasst und als Kalender verwendet. Hierbei musste die chronologische Darstellung aller Beiträge unterbunden und in der Datei ‚index.php‘ des jeweiligen THEMES assimiliert werden.²¹³

```
01  <?php
02  /// Auszug der index.php des Esperanto THEMES
03  global $query_string;
04  query_posts($query_string . "&order=ASC");
05  ?>
```

Listing 6.1: Individuelle Anpassung der Datei ‚index.php‘ des THEMES

212 Siehe: <http://wordpress.org/extend/plugins/calendar/>

213 Vgl. Mullenweg, 2010 [www-dokument] – Function Reference/query posts

Bevor der Loop beginnt, findet eine neue Abfrage der Datenbank und der `.query_post` Parameter mit Hilfe des dargestellten Codes statt. Die Reihenfolge wird durch die Anweisung `.order=ASC` umgekehrt. Um die Parameter der URL zu erhalten wird mittels der globalen Variable `.$query_string` die Information gesichert, da sie sonst ignoriert wird.²¹⁴

Ein Nachteil dieser Termindarstellung mit Hilfe der Artikelfunktion stellt sich jedoch erst im praktischen Verlauf heraus. So ist es bei nachträglich eingepflegten Terminen schwierig, diese an eine bestimmte Stelle zu positionieren. Im Backend muss innerhalb der ARTIKEL, über den ZEITSTEMPEL, der Erstellungszeitraum manipuliert werden, um einen Beitrag einzufügen. Dies ist unter anderem umständlich, da man den Zeitpunkt vom Vorgänger- und Nachfolgebeitrag kennen muss. An dieser Stelle zeigt sich, dass WordPress nicht den vollen WCMS-Umfang besitzt und nur durch ergänzende Funktionen erreicht werden kann.

6.1.1 Einbindung von sIFR

Durch die Anpassung der Artikelfunktion ist eine Integration zusätzlicher Bilder in die Termine möglich. Auch eine Bewertung der Veranstaltung mittels Kommentarfunktion wird im modifizierten Artikelmodul erzeugt. Eine besondere Funktionalität, die in der TEMPLATE-Datei `.functions.php` des THEMES integriert ist, ist das auf FLASH basierende `.Font-Replacement`-Werkzeug sIFR.

Mit dieser Technologie können individuelle Schriftarten verwendet werden, ohne dabei die Schriften in Grafiken zu rastern. Voraussetzung für den Einsatz von sIFR ist die Einbindung des Fonts in einer `.swf-Datei`, die sich auf dem Webserver befindet.²¹⁵

In WordPress kann sIFR über ein PLUGIN, eine Funktion oder direkt im Fuß der Seite eingebunden werden. Aus performancetechnischen Gründen wurde sIFR über eine Funktion in der `.functions.php` und im Footer der Datei ressourcenschonend integriert.²¹⁶

214 Vgl. Mullenweg, 2010 [www-dokument] – Function Reference/query posts

215 Vgl. Puscher, 2005 [www-dokument]

216 Vgl. Cannaday et al., 2009, [www-dokument]

```
01  <?php
02  /**
03   * @package WordPress
04   * @subpackage Esperanto_Theme
05   */
06  // sIFR
07  function sifr_js() {
08      $sifrurl = get_bloginfo('template_directory').'/sifr/';
09
10      echo '<!-- start sifr JavaScript -->
11      <script src="'. $sifrurl. 'js/sifr.js" type="text/JavaScript"></script>
12
13      <script type="text/JavaScript">
14          var test = { src: "'. $sifrurl. 'flash/europgrobol.swf" };
15      </script>
16
17      <script src="'. $sifrurl. 'js/sifr-config.js" type="text/JavaScript"></script>
18      <!-- sifr scripts -->' ;
19  }
20  function sifr_css()
21  {
22      $sifrurl = get_bloginfo('template_directory').'/sifr/';
23
24      echo '<!-- start sifr CSS -->
25      <link rel="stylesheet" href="'. $sifrurl. 'css/sifr.css" type="text/css"
26      media="screen" />
27      <!-- end sifr CSS -->';
28  }
29  add_action('wp_head', 'sifr_css');
30  add_action('wp_footer', 'sifr_js');
31  ?>
```

Listing 6.2: Implementierung von sIFR in der Datei ‚functions.php‘

In der Datei ‚functions.php‘ werden die zwei Funktionen definiert, die zum einen die CSS-Datei in den HTML-Head-Bereich einbinden und zum anderen das benötigte Javascript (‚sifr.js‘) in die Datei ‚footer.php‘ integriert. Dabei wird innerhalb der jeweiligen Funktion über die Variable (‚\$sifrurl‘) und der WordPress internen Funktion ‚get_bloginfo‘, das Verzeichnis von sIFR festgelegt,

welches über die ‚echo-Anweisung‘ die Funktionen ausgibt²¹⁷. Im letzten Abschnitt wird mittels der ACTION HOOKS bestimmt, in welchem Bereich des Dokumentes die Skripte einpflegt werden.

Das Beispiel verdeutlicht die einfache Implementierung von Funktionen und die Ausbaufähigkeit von WordPress als WCMS, wobei die Verwendung von siFR nicht zum Standardrepertoire eines WCMS gehört. Dennoch ist es möglich über Hooks eigene Skripte und Plugins einzufügen.

6.1.2 SEITEN

Die aus dem Administrationsbereich erstellten SEITEN (siehe Kapitel 4.2.5) werden in der vorgesehenen TEMPLATE-Datei („page.php“) des THEMES festgelegt. Sollte diese Datei nicht vorhanden sein, so wird automatisch die obligatorische Datei „index.php“ aufgerufen und der entsprechende Inhalt der Seite ausgegeben²¹⁸. Um eine Seitenverlinkung und somit eine Seitennavigation zu erzeugen, wurden in der Datei „header.php“ des zu entwickelnden THEMES, der TEMPLATE TAG „wp-list-pages“ eingefügt.

```
01 <ul>
02 <?php wp_list_pages(,include=2,26,28,80,29,30&sort_column=menu_order&title_li`); ?>
03 </ul>
```

Listing 6.3: Verwendung des TEMPLATE TAG „wp_list_pages“

Dieser TEMPLATE TAG ermöglicht die Basisvoraussetzung, um WordPress als seitenbasierendes Web Content Management System zu nutzen. Es wird mit Hilfe dieses TAGS eine Verlinkung zu den dynamisch erzeugten SEITEN erstellt.²¹⁹ Der TAG kann durch Angabe bestimmter Parameter wie im Beispiel sichtbar, eingegrenzt oder erweitert werden.

Im Beispiel werden die SEITEN mit der ID 2, 26, 28, 29,30 und 80 aus der Datenbank eingebunden. Der Parameter ‚sort_colum‘ sortiert die SEITEN, im dargestellten Fall nach der Eigenschaft ‚menu_order‘. Die Menu-Sortierung ‚menu_order‘ wird innerhalb des Backends bei der Erstellung festgelegt (siehe Reihenfolge, Kapitel 4.2.5). Die Sortierung kann jedoch auch andere Eigenschaften enthalten, welche den Aufteilungsprozess beispielsweise nach der ID, der letzten

217 Vgl. Cannaday et al., 2009 [www-dokument]

218 Vgl. Mullenweg, 2010 [www-dokument] – Creating Your Own Page TEMPLATES

219 Vgl. Mullenweg, 2010 [www-dokument] – Function Reference/wp list pages

Änderungen oder dem Autor bestimmt. Wenn der Parameter ‚sort_column‘ nicht registriert ist, tritt der default-Fall in Kraft²²⁰.

Um eine hierarchische Navigationsstruktur innerhalb von WordPress zu erstellen, kann der TEMPLATE TAG ‚wp_list_pages‘ mit den definierten Unterseiten aus den Backend erweitert werden. Hierfür ist es wichtig, dass der Parameter ‚child_of‘ im TEMPLATE TAG eingebunden wird. Mit dem Parameter ‚depth‘ wird festgelegt, bis in welche Ebene Unterseiten aufgelistet werden. Die Bestimmung erfolgt über ein INTEGER²²¹

Wird auf bestimmten Elternseiten (Parentpages) die Ausgabe der Kinderseiten (Childpages) als Navigationspunkte erwünscht, so kann mit dem CONDITIONAL TAG ‚is_page()‘ und einer Fallunterscheidung, die Subnavigation innerhalb der Elternseite integriert werden.

```
01  <?php
02  <?php wp_list_pages('sort_column=menu_order&depth=2&title_li='); ?>
03  if(is_page('Galery')){
04      $children = wp_list_pages('title_li=&child_of='.$post->ID.'&echo=0');
05      if($children){
06          echo '<ul>' . $children . '</ul>';
07      }
08  }
09  ?>
```

Listing 6.4: Integration von Unterseiten innerhalb des THEMES

6.1.3 Einbindung des Javascript-Frameworks JQUERY

Seit der WordPress Version 2.2 ist das Javascript-Framework JQUERY fester Bestandteil des Systems. Dieses wird hauptsächlich im Backend für eine bessere Übersicht und Strukturierung der Inhalte genutzt²²². Somit wird dem Benutzer ermöglicht, Inhalte effektiver und schneller zu verwalten. Da vor allem in modernen Webdesigns immer häufiger Wert auf ‚unobtrusives Javascript‘ gelegt wird, bietet sich die Integration des Frameworks an, um dieses im Frontend einzusetzen. So wurden im Projektbeispiel Image-Slideshows und Navigationsanimationen mit Hilfe von JQUERY realisiert. Die Implementierung der JQUERY-Kerndatei kann aus dem WordPress-Verzeichnis ‚wp-includes‘ über nachfolgende Funktion eingebunden werden.²²³

220 Vgl. Mullenweg, 2010 [www-dokument] – WP list pages Default Usage

221 Vgl. Mullenweg, 2010 [www-dokument] – WP list pages Parameters

222 Vgl. Mullenweg, 2007 [www-dokument] – Version 2.2

223 Vgl. Mullenweg, 2010 [www-dokument] – Function Reference/wp enqueue script


```

01  <?php
02  wp_enqueue_script('jquery');
03  ?>

```

Listing 6.5: Einbindung von JQUERY aus dem WordPress CORE

Sobald die Kerndatei eingebunden wurde, können eigene, sich auf JQUERY beziehende Skripte, über die Funktion ‚bloginfo‘ hinzugefügt werden. Viele WordPress-Erweiterungen nutzen ebenfalls dieses Framework, um beispielsweise Inhaltselemente mit Hilfe von TABS oder Navigationsreiterei zu unterteilen²²⁴. Hierbei muss der Nutzer bei der Erstellung von TABS innerhalb der Backend-Eingabemaske vordefinierte Schlüsselwörter in Form von festgelegten PLUGIN-TAGS verwenden, um Inhalte zu separieren.

Ein Ausschnitt aus dem Backend soll die Verwendung von JQUERY-TABS durch das PLUGIN ‚WP Tabs and Slides‘ verdeutlichen. Die gelb hinterlegten PLUGIN-TAGS sind dabei die zu erstellenden TABS. Diese werden im Frontend ausgegeben und die Inhalte durch JQUERY geteilt. Die Gestaltung der TABS kann im Frontend durch CSS angepasst werden, wobei eine Erweiterung durch IDs in der PLUGIN Datei ‚wp-tabs-slides.php‘ definiert werden muss.

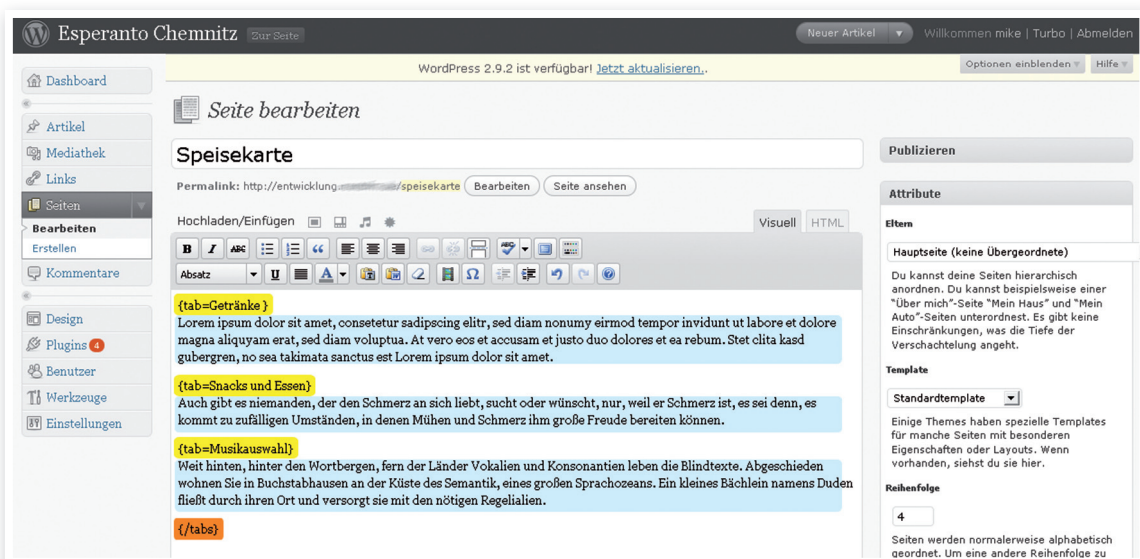


Abbildung 6.1: Nutzung des PLUGINS JQUERY TABS AND SLIDES

²²⁴ Siehe: <http://wordpress.org/extend/plugins/wordpress-tabs-slides/>

6.2 Praxisnahe WCMS-Erweiterungen für WordPress

Um das Blogsystem WordPress als Web Content Management System nutzen zu können, muss die Software mit PLUGINS oder durch eigene Skripte erweitert werden, da die Grundfunktionalitäten des Systems eingeschränkt sind. Im Kapitelabschnitt 6.1.3 wurde auf das TABS AND SLIDES PLUGIN eingegangen, welches mit Hilfe von JQUERY, Inhaltselemente in TABS teilt. In den folgenden Abschnitten werden einzelne Erweiterungen für WordPress untersucht, die für den praktischen Einsatz, innerhalb eines Web Content Management Systems, sinnvoll sein können. Viele dieser PLUGINS wurden im Praxisbeispiel erfolgreich eingesetzt.

6.2.1 NEXTGEN GALLERY

Für die Realisierung der beschriebenen Website wurden weitere PLUGINS in das System eingebunden, die den Vorstellungen des Kunden entsprachen. So ist besonders das Galerie-PLUGIN NEXTGEN GALLERY von ALEX RABE hervorzuheben. Die Erweiterung lässt sich optimal in WordPress integrieren und bietet eine Vielzahl von Möglichkeiten, die über herkömmliche Galerieanwendungen hinausgeht. Es können beispielsweise Fotoalben, Wasserzeichen und Slideshows der Galerie hinzugefügt, sowie unterschiedliche Javascript-Effekte (z.B. Thickbox, Lightbox oder Highslider) in den Galerien eingesetzt werden. Das PLUGIN erfordert vom Webserver eine Speicherkapazität von mindestens 16MB und eine PHP-Version 5.2. Für die optimale Erstellung von Vorschau Bildern (THUMBNAILS) sollte der Server mit ‚Safe Mode OFF‘ konfiguriert sein.²²⁵

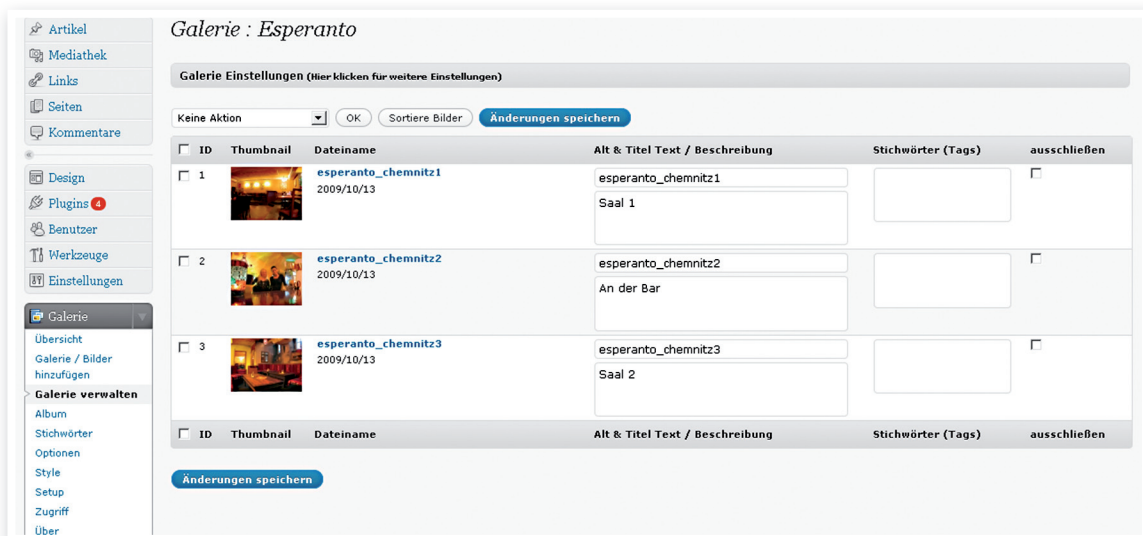


Abbildung 6.2: Erstellung von Galerien mit der Erweiterung NEXTGEN GALLERY

²²⁵ Vgl. Rabe, 2010 [www-dokument]

Nachdem die Erweiterung in das PLUGIN-Verzeichnis hochgeladen wurde, erscheint in der Administrationsoberfläche ein neuer Menüpunkt mit dem Titel GALERIE. Von hier aus können Bilder hochgeladen, betitelt und editiert werden. Die Erstellung von THUMBNAILS wird beim Hochladen des Bildformates automatisch generiert. Die jeweilige Thumbnailgröße lässt sich dabei im Untermenüpunkt OPTIONEN konfigurieren.²²⁶

Die Integration von Galerien oder Fotoalben erfolgt über die SEITEN/ARTIKEL. Es besteht die Möglichkeit den erzeugten GALERIE-TAG mit einer speziellen ID, direkt in die Seitenadministration zu integrieren oder den Vorgang über den Button NEXTGEN GALLERY im TINYMCE-EDITOR zu steuern. Eine weitere Einbindungsoption bieten widgetfähige THEMES. Innerhalb dieser THEMES kann mit Hilfe von ‚drag and drop‘ die Galerie in die SIDEBAR gezogen werden.²²⁷

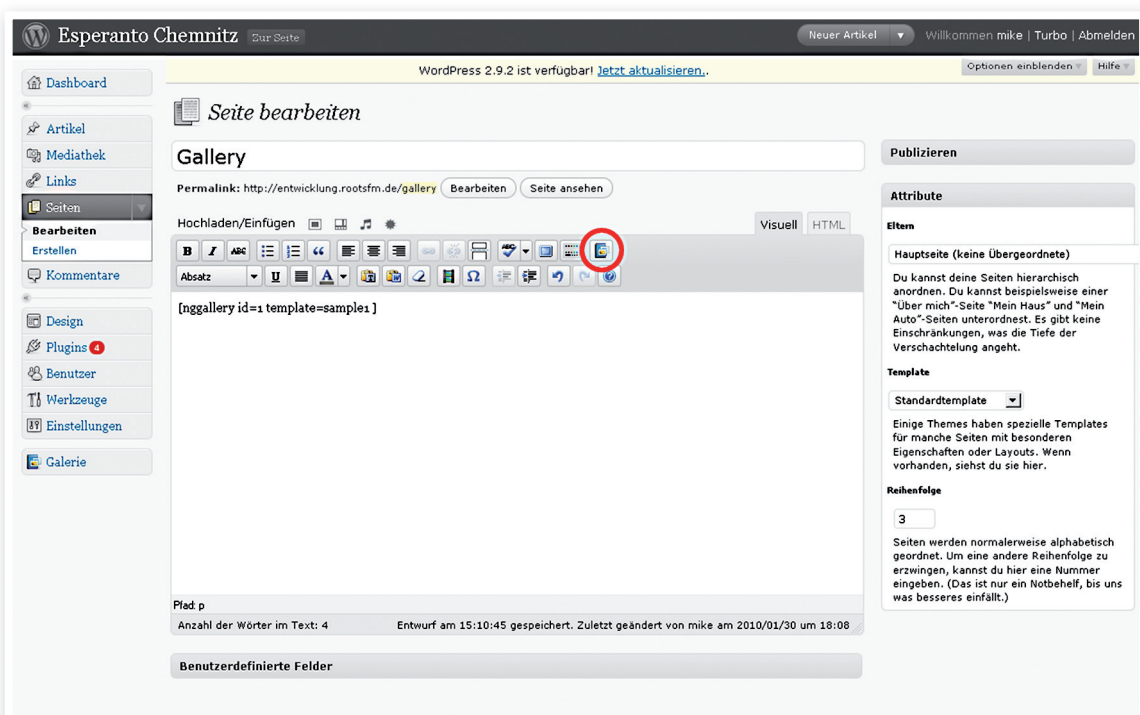


Abbildung 6.3: Einfügen einer Galerie in SEITEN

²²⁶ Vgl. Curly, 2007 [www-dokument]

²²⁷ Vgl. Schmitz (2006), [www-dokument]

6.2.2 Formulare mit CFORMS II

Die Formular-Erweiterung CFORM II von ‚deliciousdays.com‘ ermöglicht eine komfortable Implementierung von selbstdefinierten Formularen in WordPress. Nach der Installation können Formulare mit ‚Input-‘, ‚Select-‘, ‚Upload- Feldern‘, ‚Checkboxes‘, ‚Radiobuttons‘, ‚Textareas‘ und ‚Captchas‘ konfiguriert werden. Innerhalb dieser Feldtypen erlaubt das PLUGIN auch den Einsatz von ‚reguläre Ausdrücke‘ zum Validieren von Feldern. So ist es beispielsweise möglich, dass der Feldtyp ‚E-Mail‘, eine korrekte E-Mail Adresse enthält.²²⁸

Das Versenden von Formulardaten erfolgt über die, in den PLUGIN-Einstellung festgelegte, E-Mail Adresse. Alternativ können Formularinhalte in der Datenbanktabelle des PLUGINS gespeichert werden. Diese Funktionalität muss jedoch zusätzlich in der vorgesehenen Einstellung ‚Database tracking‘ vorgenommen werden.

Die Einbindung von Formularen erfolgt, ähnlich wie bei dem Galerie-PLUGIN, über die TINYMCE Schaltfläche oder als PLUGIN-Tag durch ‚<!--cforms name=“Esperanto Kontakt“-->‘. Nachdem Speichern der Seite ist die Erweiterung aktiv und wird im Frontend dargestellt.²²⁹

Feldname	Typ	required	e-mail	ajax-clear	disabled	read-only
1 Name*	Einzeiliger Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 E-Mail*	Einzeiliger Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 Nachricht*	Mehrzeiliger Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

** Add ** 1 new field(s) @ position 1

Messages, Text and Button Label

Grundlegende Formular Administrator / E-Mail Optionen

Abbildung 6.4: Ausschnitt aus der Formular-Erweiterung CFORMS II

²²⁸ Vgl. Frütel, 2009, S.165

²²⁹ Vgl. Stich, 2010, [www-dokument]

6.2.3 WP-GUESTBOOK PLUGIN

Das PLUGIN WP-GUESTBOOK von FRANK BÜLTGE basiert auf der vorhandenen Kommentarfunktion von WordPress. Es ermöglicht das Verfassen von Kommentaren, die als Gästebucheinträge im Frontend dargestellt werden. Im Backend können unter dem Button KOMMENTARE alle Gästebucheinträge verwaltet, moderiert und organisiert werden. Der Vorteil dieser Erweiterung ist vor allem, die ressourcenschonende Integration des PLUGINS in das System, da durch die Kommentarfunktion eine Schnittstelle zum CORE vorhanden ist.²³⁰

Die Konfiguration des Gästebuchs erfolgt über die Datei ‚paged-guestbook-config.php‘ im PLUGIN-Verzeichnis von WordPress. Im Gegensatz zu anderen PLUGINS, verzichtet WP-GUESTBOOK auf eine Konfigurationsschnittstelle in der Administrationsoberfläche. Innerhalb der Konfigurationsdatei kann die Reihenfolge, die maximalen Einträge pro Seite und der Seitenumfang bestimmt werden. Die zu bestimmenden Einstellungen werden in den vordefinierten PHP-Variablen angepasst und gesichert. Sobald die Datei erneut auf den Webserver hochgeladen wurde, treten die definierten Eigenschaften in Kraft.²³¹

Die Implementierung des Gästebuchs erfolgt in der zuvor erstellten Seite ‚Gästebuch‘. Das PLUGIN differenziert sich gegenüber den beiden Erweiterungen ‚NextGen Gallery‘ und ‚Cforms II‘ in der Einbindung. So wird das PLUGIN nicht in den Inhaltsbereich mittels PLUGIN-Tags integriert, sondern über eine spezielle TEMPLATEDATEI aus dem Theme-Verzeichnis. Dieses muss sich im Theme-Ordner befinden und innerhalb des LOOPS folgenden Code aufrufen:

```
01      <?php if (function_exists('paged_guestbookcomments_template')) paged_
guestbookcomments_template();
02          else
03              if (function_exists('guestbook_template')) guestbook_template(); else
04                  comments_template();
05      ?>
```

Listing 6.6: Auszug aus der Datei: ‚wp-guestbook_page.php‘²³²

230 Vgl. Bültge, 2006, [www-dokument] - WP - Gästebuch [WP-Guestbook] (PLUGIN)

231 Vgl. Bültge, 2006, [www-dokument] - WP - Gästebuch [WP-Guestbook] (PLUGIN)

232 Vgl. Bültge, 2006, [www-dokument] - WP - Gästebuch [WP-Guestbook] (PLUGIN)

Sämtliche Formatierungen und Anpassungen des Gästebuchs erfolgen in der TEMPLATE-DATEI ‚wp-paged-guestbook.php‘. Diese Datei ist ähnlich strukturiert wie ‚comments.php‘, unterscheidet sich jedoch in einigen Details, wie beispielsweise die Vergabe einer Seitennummerierung oder eines Spamschutzes.²³³

```
01  <!-- WP-Paged-Guestbook page numbers -->
02  <hr class="dotted" />
03  <?php if ($paged_guestbookcomments->pager->num_pages() > 1): ?>
04  <p class="comment-page-numbers">
05      <?php paged_guestbookcomments_print_pages(); ?>
06  </p>
07  <?php endif; ?>
08  <!-- End WP-Paged-Guestbook page numbers -->
```

Listing 6.7: Einbindung einer Paginierung in der Datei ‚wp-paged-guestbook.php‘²³⁴

6.2.4 Suchmaschinenoptimierung mit WP-ALL IN ONE SEO

WordPress unterstützt in der Grundfunktion suchmaschinenfreundliche URLs in Form von PERMALINKS. Da diese Ausstattung für SEO nicht genügt, kann mit dem PLUGIN ALL IN ONE SEO ein verbessertes Ranking bei Suchmaschinen erreicht werden²³⁵. Das PLUGIN erweitert WordPress mit artikel- und seitenbezogenen METATAGS und ermöglicht es, Titel- und Seitenbeschreibungen vorzunehmen.²³⁶

Die Installation erfolgt, nachdem das PLUGIN auf den Webserver hochgeladen wurde, über das Backend. Nach der Aktivierung werden sämtliche Konfigurationen im Menüpunkt EINSTELLUNGEN – ALL IN ONE SEO PACK vorgenommen. Von hier aus können globale Anpassungen wie z.B. der Haupttitel, eine Beschreibung der Website oder die wichtigsten Schlüsselwörter angelegt werden.²³⁷

233 Vgl. Bültge, 2006, [www-dokument] - WP - Gästebuch [WP-Guestbook] (PLUGIN)

234 Vgl. Bültge, 2006, [www-dokument] - WP - Gästebuch [WP-Guestbook] (PLUGIN)

235 Vgl. Belik, 2010, S. 301

236 Siehe: <http://wordpress.org/extend/plugins/all-in-one-seo-pack/>

237 Vgl. Torbert, 2010, [www-dokument]

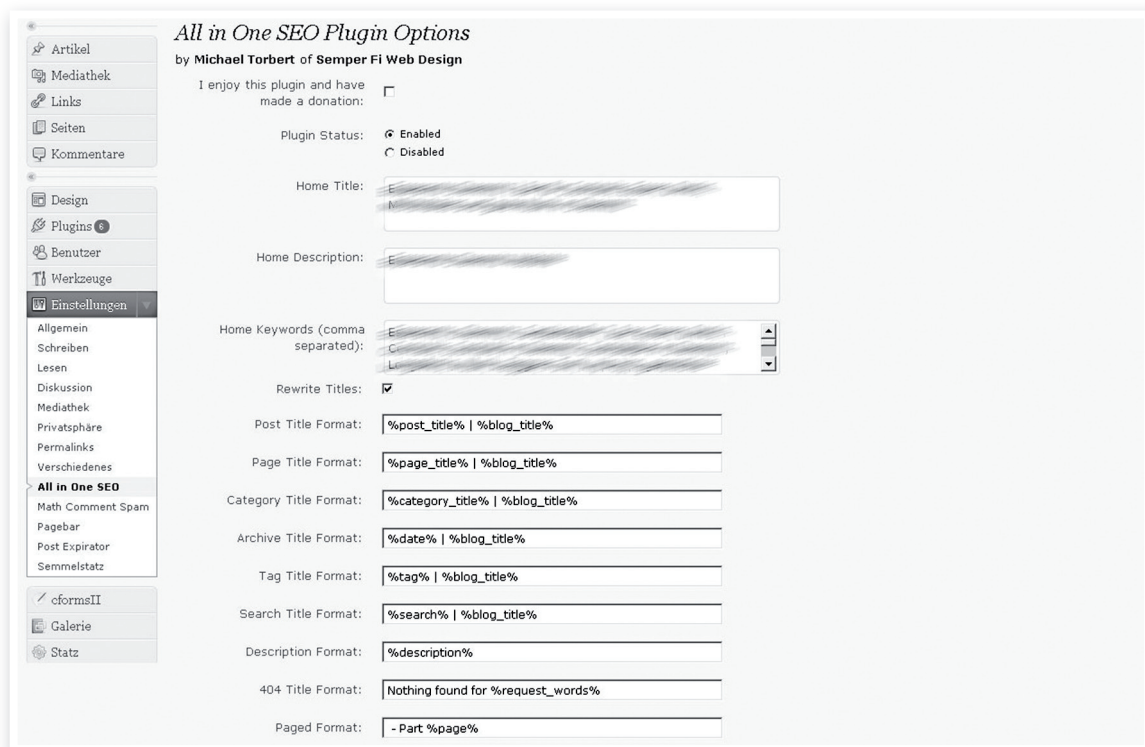


Abbildung 6.5: PLUGIN Einstellung für ALL IN ONE SEO

Eine detaillierte Vergabe von Titeln, Schlüsselwörter, Beschreibungen und Titelattributen kann direkt in den jeweiligen SEITEN und ARTIKELN der Administrationsoberfläche erstellt werden. Das PLUGIN bietet darüber hinaus Hinweise, wie viele Suchbegriffe von Suchmaschinen initialisiert werden, sodass auch Laien die Pflege übernehmen können.

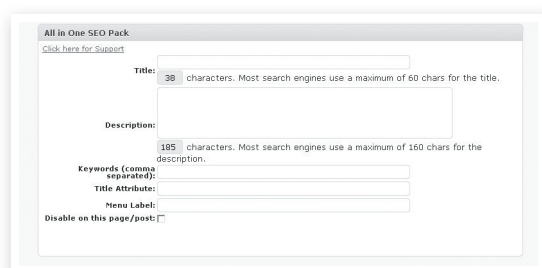


Abbildung 6.6: Vergabe von Schlüsselwörter in SEITEN mit Hilfe von ALL IN ONE SEO

6.2.5 Weitere praxisrelevante WCMS-Erweiterung für WordPress

Für eine optimale Nutzung von WordPress als Web Content Management System eignen sich in der Praxis auch nachfolgende Erweiterungen:

DB-BACKUP:

Das PLUGIN DB-BACKUP ermöglicht die einfache Sicherung aller Datenbankeinträge und Datenbanktabellen von WordPress. Es lassen sich mit dieser Erweiterung automatische Sicherungen erstellen, die auf dem Server oder dem lokalen Rechner gespeichert werden. Außerdem bietet das PLUGIN die Option an, „SQL-Dumps“ per E-Mail zu versenden.²³⁸

WP SEMMELSATZ:

SEMMELSATZ ist eine Erweiterung für WordPress zur Erfassung von Besucherstatistiken. Das PLUGIN listet im Backend sämtliche Besucherströme und verdeutlicht diese mit grafischen Diagrammen. Es werden die neuesten Suchbegriffe („aktuelle ‚Referer‘“), die beliebtesten Seiten und Beiträge, sowie Besucherentwicklungen zeitlich dargestellt²³⁹. Die Weiterentwicklung des PLUGINS wurde jedoch eingestellt, sodass für zukünftige WordPress-Veröffentlichungen nur der jetzige Funktionsumfang des PLUGINS zum Einsatz kommen kann. Eine Alternative bietet das PLUGIN WP-STATS.²⁴⁰

PAGEBAR:

Die Erweiterung PAGEBAR ermöglicht die Einbindung einer ‚Pagination‘ für ARTIKEL und KOMMENTARE. Somit lassen sich ARTIKEL und KOMMENTARE auf beliebig viele Unterseiten aufteilen und separieren. Das PLUGIN optimiert die Darstellung der Seite und teilt Inhaltselemente layoutgerecht auf. PAGEBAR ist vor allem bei Webseiten vorteilhaft, bei denen häufig Neuigkeiten in Form von ARTIKELN publiziert werden.²⁴¹

WP-POLL:

WP-POLL ist eine spezielle Umfrageerweiterung für WordPress. Durch dieses PLUGIN lassen sich einfache Umfragen mit benutzerdefinierten Fragen erstellen. WP-POLL ist durch TEMPLATES und CSS anpassbar und bietet innerhalb von Umfragen die Option, Mehrfachauswahlen zu erzeugen.²⁴² Die Erweiterung ist eine Alternative zu WP-POLLDADDY, welche einen ähnlichen Funktionsumfang bietet, jedoch erst unter einer PLUGIN-Domain registriert werden muss.²⁴³

²³⁸ Vgl. Frütel, 2009, S. 137f.

²³⁹ Vgl. Piringer, 2009, S.13

²⁴⁰ Siehe: <http://wordpress.org/extend/plugins/wp-stats/>

²⁴¹ Vgl. Schröder, 2010 [www-dokument]

²⁴² Vgl. Chan Hong Wie, 2010 [www-dokument]

²⁴³ Vgl. Frütel, 2009, S. Seite 161f.

POST-EXPIRATOR

Dieses PLUGIN erzeugt einen festen Ablauf-Zeitpunkt von ARTIKEL und SEITEN. Nach dem Ablauf des definierten Zeitpunktes, werden ARTIKEL oder SEITEN entweder direkt gelöscht oder im Backend zu ENTWÜRFEN verlegt. Innerhalb von WordPress können Veröffentlichungszeitpunkte definiert werden, jedoch keine Ablaufzeiten²⁴⁴.

ROLE MANAGER

Mit eines der wichtigsten Aufgaben in einen WCMS ist die Nutzer- und Rollenverwaltung. WordPress bietet in der Grundarchitektur ein eingeschränktes Nutzerverwaltungssystem. Mit Hilfe des PLUGINS ROLE MANAGER lässt sich dieses Verwaltungssystem ausbauen und erweitern. Es können explizite Tätigkeiten dem Benutzer zugeordnet werden, ohne das dies globale Auswirkung auf alle anderen Benutzer hat.²⁴⁵ Eine komplexere und detailliertere Benutzer- und Rollenverwaltung lässt sich mit der Erweiterung ROLESCOPER realisieren.²⁴⁶

WPML MULTILINGUAL CMS

Diese Erweiterung ermöglicht den Einsatz und die Verwaltung mehrerer Sprachen in WordPress. In Form von Reitern oder als Dropdown Menü können in der Administrationsoberfläche unterschiedlichste Übersetzungen separat erstellt werden²⁴⁷. Das PLUGIN eignet sich vor allem für eine internationale Präsentation, bei denen eine Vielzahl von Sprachen Verwendung finden. Des Weiteren bieten die Hersteller der Erweiterung eine professionelle, kostenpflichtige Übersetzung des jeweiligen Webinhalts an.²⁴⁸

6.3 Nachteile von WordPress als WCMS

Gerade praktische Erfahrungsberichte von Nutzern zeigen, neben der komfortablen Erweiterbarkeit von WordPress als WCMS, Realisierung-Probleme von Projekten. So wird beispielsweise das Backend bei einer großen Anzahl an verschachtelten SEITEN unübersichtlich und verliert gegebenenfalls an Performance. Auch die Verwendung einer hohen Anzahl an Erweiterungen wirken sich negativ auf die Geschwindigkeit des Systems aus²⁴⁹. In einer neuinstallierten WordPress-Testumgebung kommt es zu 17 Datenbank-Abfragen im WordPress DEFAULT-THEME. Mit mehreren PLUGINS kann die Anzahl der Abfrage schnell auf 40 oder mehr steigen.

²⁴⁴ Vgl. Adler, 2009, S. 201

²⁴⁵ Vgl. Schneider, 2007 [www-dokument]

²⁴⁶ Vgl. Behrens, 2010 [www-dokument]

²⁴⁷ Vgl. OnTheGoSystems, Inc., 2010 [www-dokument]

²⁴⁸ Vgl. Petereit, 2009 [www-dokument]

²⁴⁹ Vgl. Frütel, 2009, S. 256

An dieser Stelle verdeutlicht das System auch den eigentlichen Ursprung. WordPress wurde von den Herausgebern als Blogsystem entwickelt. Viele im Core vorhandene Eigenschaften, wie zum Beispiel TRACKBACKS, PINGBACKS oder KOMMENTARE werden oftmals in der Praxis nicht als WCMS-Feature eingesetzt. Sie dienen vorrangig der Kommunikation von Blogs und deren Lesern²⁵⁰. WordPress ist somit oftmals im praktischen Einsatz zu überdimensioniert oder reicht für gängige WCMS-Eigenschaften nur bedingt aus und muss durch evtl. unsichere PLUGINS nachgerüstet werden.

Die Thematik Sicherheit spielt gerade bei externen Erweiterungen oder THEMES eine wesentliche Rolle. Da WordPress sich durch offenen Quellcode auszeichnet und somit jeder eigene PLUGINS und THEMES erstellen kann, entsteht ein erhöhtes Sicherheitsrisiko. Aus diesem Grund empfiehlt es sich, nur registrierte und bekannte Erweiterungen von der Betreiberplattform zu beziehen, um schädlichen Code zu vermeiden²⁵¹. Neue PLUGINS und THEMES sollten aus diesem Grund immer erst lokal getestet werden, bevor sie im Live-System zum Einsatz kommen.

Weitere Sicherheitsrisiken verbergen sich auch im Login-File (.wp-login.php) von WordPress. Nichtbefugten Personen ist es vom Kern-System gestattet, unendlich viele Login-Versuche zu unternehmen, ohne das jeweilige Konto oder Benutzer zu sperren. Ein Angreifer kann durch gezielte Algorithmen sich somit Zugang zum System verschaffen. Standardmäßig ist der Benutzername für den Administrator des System als ADMIN konfiguriert. Wird der Benutzername des Administrators nicht geändert, bietet dies für Hacker einen weiteren Angriffspunkt durch Brute-Force Attacken.²⁵²

Ein nächstes Defizit beim Einsatz von WordPress als WCMS spiegelt sich in der Rollen- und Rechtevergabe wider. Die Verwaltung und Zuweisung von Rechten, können mit den beschriebenen PLUGINS ausreichend erweitert werden, jedoch gibt es häufig Probleme bei der Bearbeitung von SEITEN oder ARTIKELN, wenn diese von mehreren Personen editiert werden. So weist WordPress den Nutzer nicht explizit darauf hin, dass der jeweilige ARTIKEL / SEITE von einem anderen Nutzer bereits bearbeitet wird.²⁵³

Ebenso bei der Ausgabe von benutzerspezifischen Inhalten verdeutlicht WordPress WCMS-Schwächen. So ist es beispielsweise für REGISTRIERTE LESER oder ausgewählte Nutzer nicht möglich, erweiterte und geschützte Inhalte zu erhalten. WordPress bietet im Kernsystem einen Passwortschutz für SEITEN und ARTIKEL an, jedoch existieren diese Inhaltstypen auch für normale Besucher im Frontend und werden durch die Eingabe eines Passwortschutz versteckt. Eine optimale Lösung wäre die Unkenntlichkeit geschützter Seiten im Frontend.²⁵⁴

²⁵⁰ Vgl. Maurice, 2007, S. 134

²⁵¹ Vgl. Adler, 2009, S. 241f.

²⁵² Vgl. Bülgte, 2009, S. 134

²⁵³ Vgl. Adler, 2009, S. 201

²⁵⁴ Vgl. Adler, 2009, S. 201 f.

7. Vergleich zwischen TYPOLIGHT und WordPress

Die Untersuchung im vorangegangenen Kapitelabschnitt 6.3 stellte bereits einige Nachteile von WordPress als Web Content Management System dar. Im Kapitel 7 sollen weitere Vor- und Nachteile von WordPress und dessen Nutzbarkeit als WCMS erarbeitet werden. Hierzu eignet sich ein direkter Vergleich zwischen dem freien Open Source CMS TYPOLIGHT und der Blogsoftware WordPress.

7.1 Kurze Einführung zum Open Source WCMS TYPOLIGHT

Der Hauptentwickler und Herausgeber, LEO FEYER, veröffentlichte das Web Content Management System TYPOLIGHT erstmals im Jahr 2006. TYPOLIGHT basiert ebenfalls wie WordPress auf der Programmiersprache PHP und einer SQL-Datenbank²⁵⁵. Trotz wachsender Popularität kam es immer häufiger zu Verwechslungen mit dem Open Source WCMS TYPO3. Aus diesem Grund entschied sich das Entwickler-Team am 15. Mai 2010, TYPOLIGHT umzubenennen und ab der Version 2.9 unter dem Namen Contao zu veröffentlichen.²⁵⁶

TYPOLIGHT bietet durch den modernen Programmaufbau einen hohen Sicherheitsstandard und ist durch integrierte Module sowie externe Erweiterung enorm ausbaufähig. Die Flexibilität des Systems umfasst neben dem browserübergreifenden CSS-Framework und der Generierung suchmaschinenfreundliche URLs ebenfalls eine umfangreiche Rechte- und Benutzerverwaltung. Das System enthält bereits im Kern viele Funktionen, wie beispielsweise die Seiten- und Layoutverwaltung, News und Kommentarfunktion, einen Eventkalender, Formulargeneratoren, FAQ- und Glossarschnittstellen, sowie Galerie- und Newslettermodule.²⁵⁷

7.2 Ausgewählte WCMS-Eigenschaften im Vergleich

TYPOLIGHT ist ähnlich wie WordPress in zwei Bereiche, dem Frontend und Backend, gegliedert. Der Administrationsbereich ist nach der erfolgreichen Installation unter der jeweiligen Domain mit dem Anhang `./typolight` erreichbar²⁵⁸. Bereits während des Installationsvorganges differenzieren sich WordPress und TYPOLIGHT in einigen Eckpunkten.

²⁵⁵ Vgl. Feyer, 2010, S. 15ff.

²⁵⁶ Vgl. Feyer, 2010, [www-dokument] - Aus TYPOLight wird Contao

²⁵⁷ Siehe: <http://www.typolight.org/>

²⁵⁸ Vgl. Feyer, 2010, S. 51

7.2.1 Installation

Die Installation von TYPOLIGHT erfolgt mit dem Entpacken und Hochladen des File-Systems auf dem Webserver. Nach diesem Vorgang wird ähnlich wie bei WordPress das Installationstool aufgerufen, welches unter der jeweiligen Domain mit dem Anhang `./typolight/install.php` erreichbar ist. Um den Anforderungen an das System gerecht zu werden, bietet es sich an, dass von TYPOLIGHT bereitgestellte Diagnose-Tool einzusehen. Hier werden alle Servereinstellungen und Voraussetzungen für das System überprüft und aufgelistet. TYPOLIGHT benötigt mindestens die PHP-Version 5.2 und MySQL 4.1.²⁵⁹

Nachdem die FTP-Daten in das Installationstool eingegeben wurden, folgt die erste Sicherheitsbestimmung von TYPOLIGHT. Das System fordert für die Installation ein eigenständiges Passwort. Dieses Passwort ist lediglich für den Installations- und Updatevorgang bestimmt und hat keine Bedeutung für das spätere Administrationspasswort. Auf diese zusätzliche Sicherheitsmaßnahme verzichtet WordPress vollständig. Das Installtool ist außerdem gegen Brute-Force-Attacken geschützt und wird automatisch gesperrt wenn mehr als drei fehlerhafte Login-Versuche unternommen werden, wobei die Aufhebung dieser Sperre nur durch die Systemdatei (`.localconfig.php`) manuell erfolgen kann²⁶⁰. Nachdem das Installationspasswort bestimmt wurde, muss es erneut zur Verifizierung eingegeben werden.

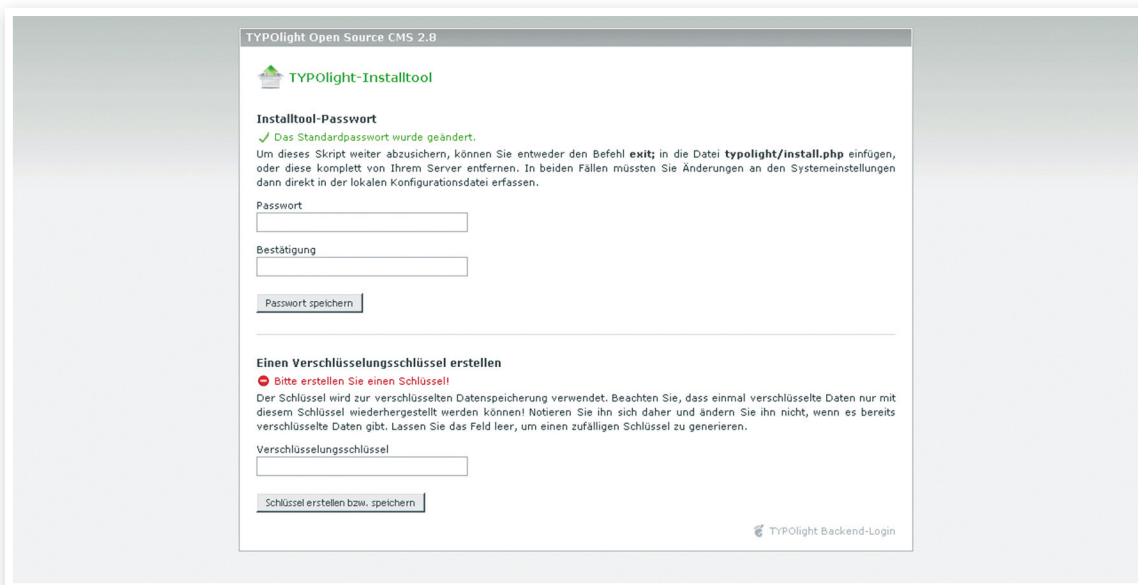


Abbildung 7.1: Installtool von TYPOLIGHT

²⁵⁹ Vgl. Feyer, 2010, S. 26

²⁶⁰ Vgl. Feyer, 2010, S. 37

Im nächsten Schritt der Installation wird ein Encryption-Key für die Verschlüsselung von Daten erstellt. Auch diese zusätzliche Sicherheitsstufe findet man in WORDPRESS nicht. Stattdessen wird bei einer WORDPRESS-Installation lediglich das Administrationspasswort automatisch generiert und per E-Mail dem Administrator zugestellt. Diese fragwürdige und unsichere Methode umgeht TYPOLIGHT ebenso, indem der Benutzername und das Passwort für den Administrator vom Installierenden selbstständig festgelegt werden. In den Grundeinstellung existiert somit kein Benutzername ADMIN, dessen Passwort durch Brute-Force-Attacken ermittelbar ist.

7.2.2 Administrationsoberflächen

Die bereits im Installtool analysierten Sicherheitsaspekte spiegeln sich auch im Login-Bereich von TYPOLIGHT wider. Sollte beispielsweise ein Nutzer dreimal hintereinander eine falsche Benutzer-Passwortkombination eingegeben haben, so wird das Konto für fünf Minuten gesperrt. Bei WORDPRESS dagegen ist die Eingabe ohne weiteres Sicherheits-PLUGIN nicht begrenzt.²⁶¹

Innerhalb des Backends ähneln sich beide Systeme vom Grundaufbau. TYPOLIGHT verfügt wie WORDPRESS über ein DASHBOARD. Dieses ist im Vergleich zu WORDPRESS jedoch auf interne Inhalte und Leistungen beschränkt und ermöglicht keine Einbindung von FEEDS oder Neuigkeiten der systembezogene WCMS-Umgebung - ausgenommen sind Informationen über Updates²⁶². In beiden Systemen können die DASHBOARDS mit PLUGINS erweitert werden²⁶³. Vorteilhaft bei TYPOLIGHT ist vor allem die Information über Tastaturkürzel für das schnelle Bearbeiten von Inhalten. Bei WORDPRESS dagegen zeichnet sich das DASHBOARD über QUICKPRESS, dem schnellen Erstellen von Nachrichten, aus.



Abbildung 7.2: DASHBOARD von WORDPRESS

²⁶¹ Vgl. Eenfeldt, 2009 [www-dokument]

²⁶² Vgl. Feyer, 2010, S. 52

²⁶³ Siehe: <http://www.viper007bond.com/wordpress-plugins/discontinued-plugins/dashboard-widget-manager/> und <http://www.typolight.org/erweiterungsliste/view/dashboard.10020009.de.html>

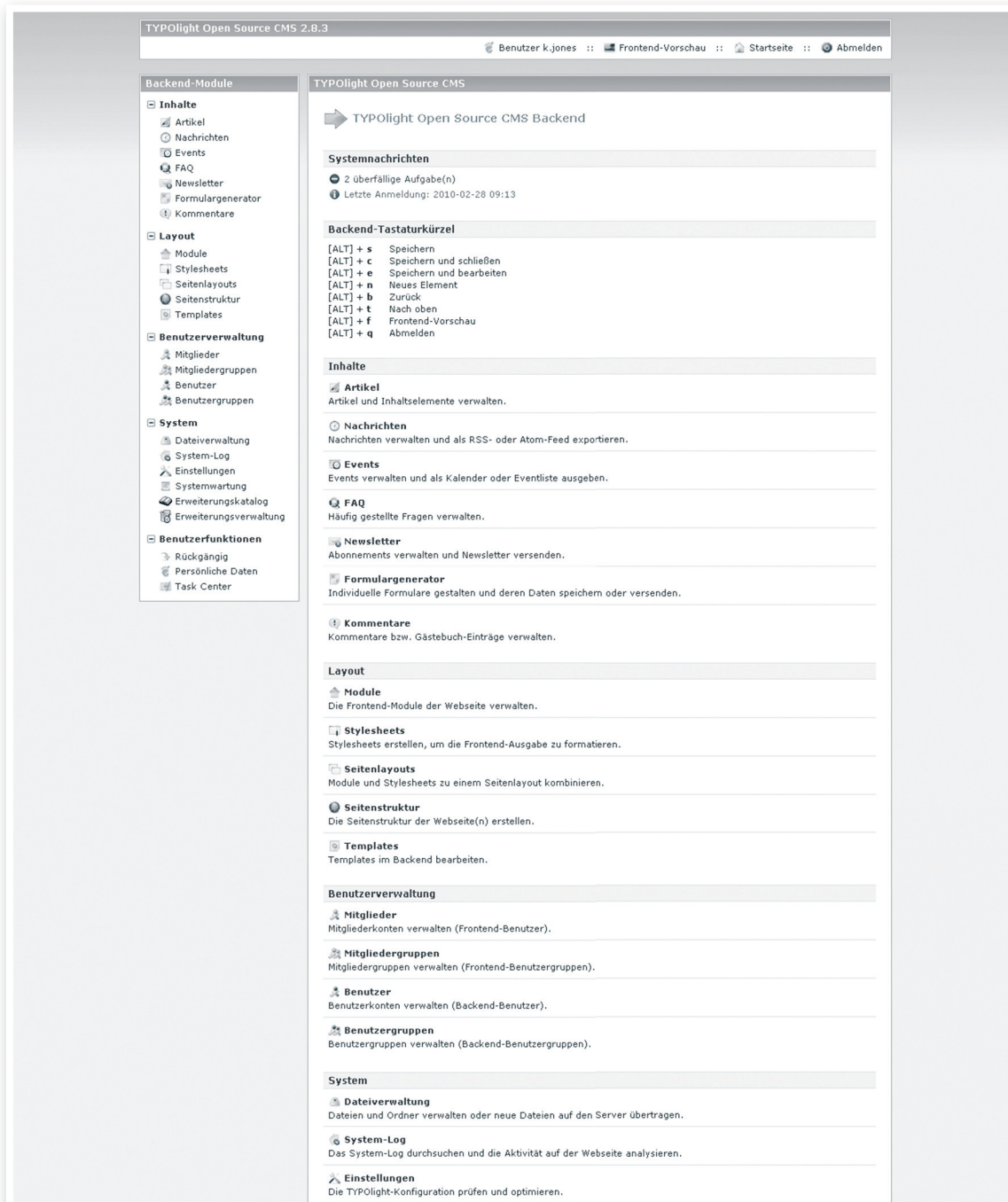


Abbildung 7.3: DASHBOARD von TYPOLIGHT

Ein wesentlicher Unterschied im Vergleich beider Systeme wird bereits zu Beginn eines Publikationsprozesses deutlich. Während WORDPRESS mit zwei unterschiedlichen THEMES ausgestattet ist und somit die Grundlage für eine Frontend-Vorlage schafft, wird TYPOLIGHT bis zur Version 2.8 ohne Theme- und Designverwaltung veröffentlicht. Lediglich ein Beispielttemplate lässt sich während der Installation integrieren.²⁶⁴

WORDPRESS ermöglicht nach der Installation ein sofortiges Publizieren von Inhalten. Des Weiteren stehen für das System über 1200 freie THEME-Vorlagen auf der offiziellen Website zum Download zur Verfügung²⁶⁵. Diese sind neben der einfachen Benutzeroberfläche auch die zwei Hauptaspekte für den enormen Erfolg von WORDPRESS.²⁶⁶

In TYPOLIGHT muss eine Vielzahl an Konfigurationen und Einstellungen vorgenommen werden, um Inhalte in ein entsprechendes Layout zu veröffentlichen. So müssen zum Beispiel das Seitenlayout mit Stylesheets, Seiten in der Seitenstruktur, Module in der Modulverwaltung, Artikel und Inhaltselemente angelegt werden, um die ersten Inhalte im Frontend darzustellen.

Die folgenden Grafiken verdeutlichen im Allgemeinen die Arbeitsphilosophie beider Systeme. Es wird ersichtlich, dass TYPOLIGHT wesentlich höhere Ansprüche an eine WCMS-Verwaltung stellt als WORDPRESS. Vor allem die Unterteilung verschiedener Inhaltselemente, fehlt bei WORDPRESS vollständig. Somit können beispielsweise SEITEN in WORDPRESS ausschließlich mit dem WYSIWYG-Editor oder dem Quellcode-Editor bearbeitet werden. In TYPOLIGHT lassen sich dagegen unendlich viele Inhaltselemente pro Seite innerhalb der ARTIKEL, EVENTS, NACHRICHTEN etc. erzeugen, womit sich der Inhalt einer SEITE optimaler verwalten bzw. organisieren lässt.²⁶⁷

²⁶⁴ Vgl. Metzmacher, 2010 [www-dokument]

²⁶⁵ Siehe: <http://wordpress.org/extend/themes/>

²⁶⁶ Vgl. Simovic et al., 2010, S. 39

²⁶⁷ Vgl. Feyer, 2010, S. 98

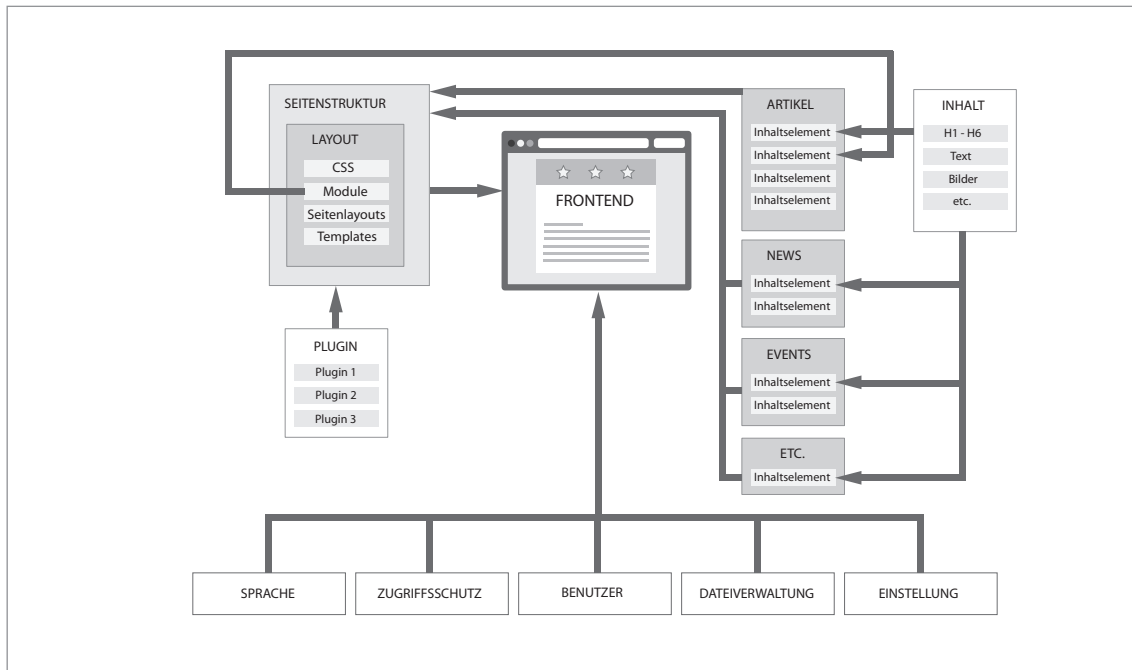


Abbildung 7.4: Systemarchitektur von TypoLight, eigene Darstellung

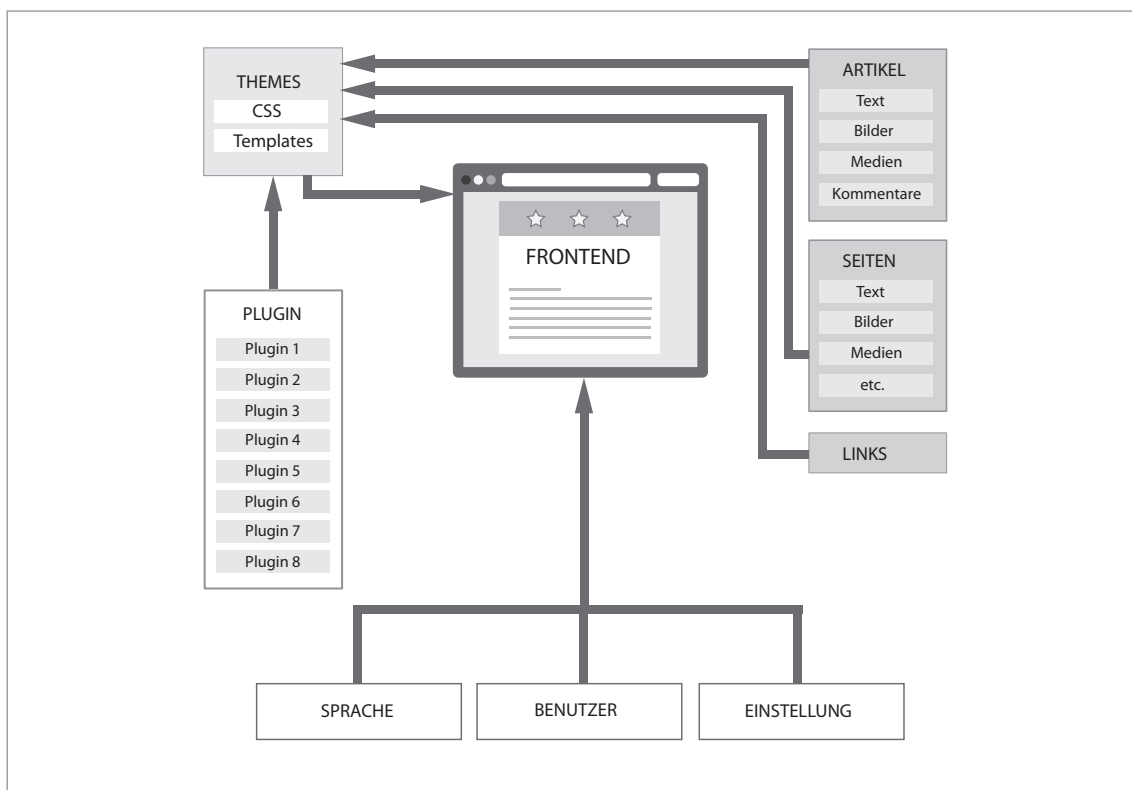


Abbildung 7.5: Systemarchitektur von WordPress, eigene Darstellung

7.2.3 Vorteile von TYPOLIGHT-Inhaltselemente gegenüber WORDPRESS-Seiten

Durch die Inhaltselemente von TYPOLIGHT lassen sich Inhalte von Seiten in mehrere Teile aufspalten. Das hat vor allem bei großen Textpassagen den Vorteil, dass diese sich übersichtlicher verwalten lassen als in WORDPRESS. Dies ermöglicht eine verbesserte Formatierung und Strukturierung der Seite. Des Weiteren können Inhaltselemente beliebig in der Hierarchie geordnet, bearbeitet, dupliziert oder gelöscht werden, womit der Inhalt flexibel bleibt und eine erneute Formatierung überflüssig ist.²⁶⁸

Im Gegensatz zu TYPOLIGHT arrangiert WORDPRESS die Einbindung von Inhalten lediglich in einem Inhaltselement – der eigentlichen Seite. Damit kann der Nutzer bei großen Textmengen den Überblick über bereits getroffene Formatierung verlieren und Inhalte nicht flexibel genug anpassen. Für kurze Inhalte ist diese Lösung jedoch ausreichend.

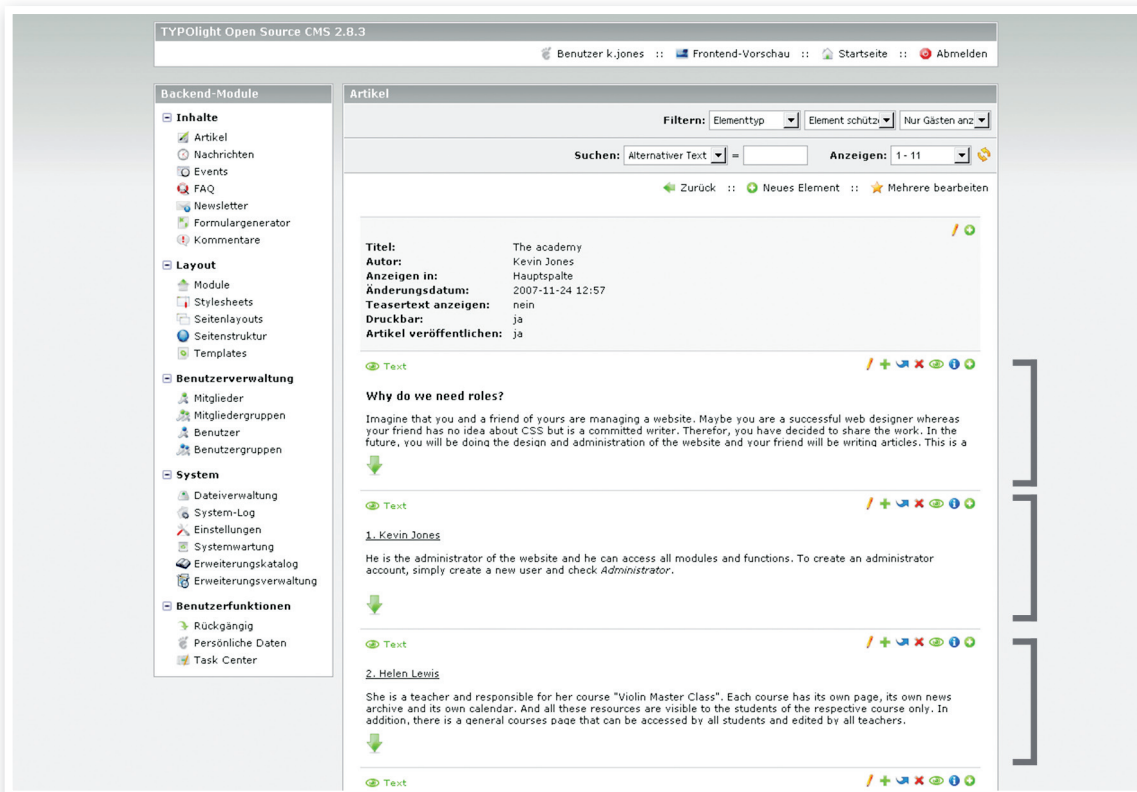


Abbildung 7.6: TYPOLIGHT Inhaltselemente

²⁶⁸ Vgl. Feyer, 2010, S. 98

7.2.4 Vorteile von TYPOLIGHT-MODULEN gegenüber WORDPRESS-WIDGETS

Module sind in TYPOLIGHT die eigentlichen Funktionsträger. Man unterscheidet grundsätzlich zwei Modultypen, den Frontend- und Backendmodulen. Frontendmodule können innerhalb der SEITENSTRUKTUR im SEITENLAYOUT, in ARTIKELN, NACHRICHTEN, EVENTS, FORMULAREN etc. eingebunden werden und stellen sämtliche Konfigurationen und Inhalte für das Frontend dar. Backendmodule dagegen sind Vermittler und Bereitsteller für Inhaltsoberflächen, in denen bestimmte Werte oder Attribute eingegeben und in der Datenbank gespeichert werden. Somit lassen sich mit Backendmodulen neue Funktionen in TYPOLIGHT, wie beispielsweise BENUTZERDEFINIERTEN SCHNITTSTELLEN, in das System implementieren.²⁶⁹

Der Vorteil dieser Modularisierung liegt ähnlich wie bei den TYPOLIGHT-INHALTSELEMENTEN in der Flexibilität. Module lassen sich (fast) überall in TYPOLIGHT integrieren und stellen der Website neue Funktionen und Inhalte zur Verfügung. Diese Funktionalität gibt es in WORDPRESS nur in eingeschränkter Form durch WIDGETS. WIDGETS bilden in der Regel eine Schnittstelle zwischen WORDPRESS-PLUGIN und THEME, welches die Flexibilität des Systems einschränkt. Somit ist es beispielsweise nicht möglich, dem System mitzuteilen, dass die drei letzten Kommentare im Fußbereich der Seite stehen sollen, die neusten Nachrichten im Kopfbereich, Bildergalerien in der SIDEBAR und im Inhaltsbereich. Nur durch manuelle Anpassung der THEMES sind diese Anforderungen möglich²⁷⁰. In TYPOLIGHT hingegen lassen sich durch die Modularisierung sämtliche Funktionen über das Backend konfigurieren.

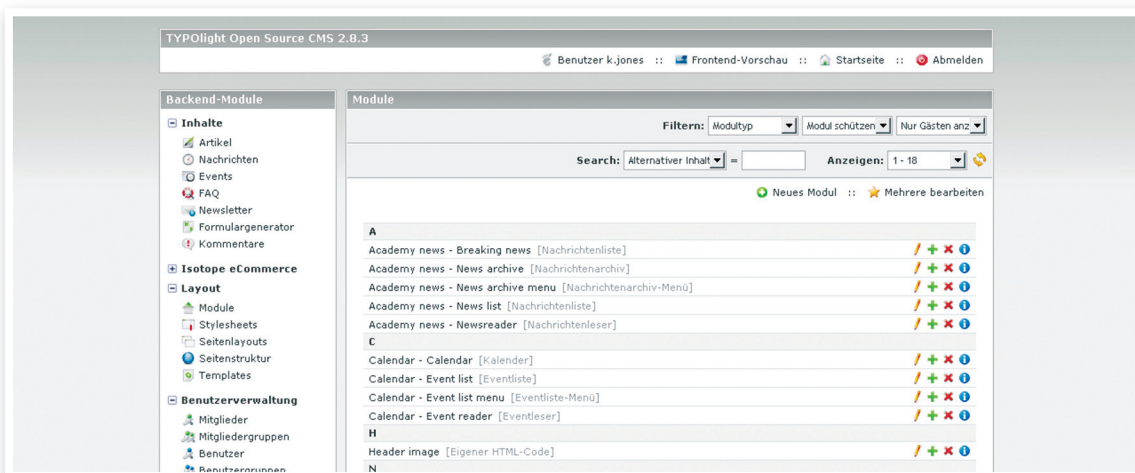


Abbildung 7.7: Modulverwaltung von TYPOLIGHT

²⁶⁹ Vgl. Feyer, 2010, S. 143 ff.

²⁷⁰ Vgl. Meyer-Gellersen, 2006 [www-dokument]

7.2.5 Sortierung und Filterung von Datensätzen im Backend

Im Kapitelabschnitt 6.3 wurden bereits die Ordnungsprinzipien von SEITEN und ARTIKEL in WORDPRESS erörtert. Bei einer hohen Anzahl an verschachtelten SEITEN und vielen ARTIKEL entsteht eine gewisse Unübersichtlichkeit im System, welches sich bei der Verwaltung von Inhalten negativ auswirkt. Die Filter- und Sortierfunktion von WORDPRESS sind häufig eingeschränkt, um mehrere Inhalte optimal zu organisieren. Im Gegensatz dazu ermöglicht TYPOLIGHT eine explizite Filterung und Sortierung von Beiträgen, sowie das Hinzufügen eigener Suchwörter für einen speziellen Suchkontext. Auch das Aus- und Einklappen verschachtelter SEITEN oder ARTIKEL sorgt für eine übersichtlichere Darstellung als in WORDPRESS.²⁷¹

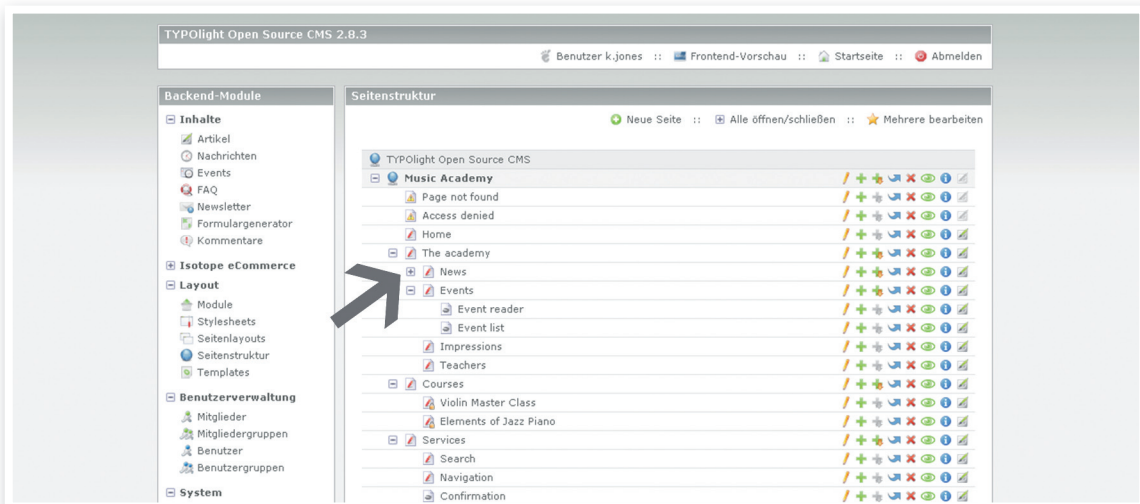


Abbildung 7.8: Auf- und Einklappen von SEITEN in TYPOLIGHT

7.2.6 Mehrfachbearbeitung

Eine weitere, komfortablere Begünstigung von TYPOLIGHT im Vergleich zu WORDPRESS ist die Mehrfachbearbeitung von Datensätzen. In WORDPRESS muss jede SEITE nacheinander bearbeiten werden, um adäquate Inhalte auszutauschen. TYPOLIGHT ermöglicht dagegen die Bearbeitung von mehreren SEITEN, ARTIKELN und INHALTSELEMENTE. Durch klicken auf den ‚Link‘ MEHRERE BEARBEITEN werden die Navigationssymbole durch Checkboxes ersetzt, mit denen sich die Datensätze auswählen lassen. Im Folgeschritt ermöglicht eine explizite Auswahl, welche Eingabefelder editiert werden können.²⁷²

²⁷¹ Vgl. Feyer, 2010, S. 77

²⁷² Vgl. Feyer, 2010, S. 63

7.3 Dateiverwaltung

Die Dateiverwaltungen in TYPOLIGHT und WordPress sind im Aufbau relativ ähnlich. So werden in WordPress alle Dokumente, wie zum Beispiel Bilder, Videos, pdf-Dateien etc. in der MEDIATHEK abgelegt und verwaltet²⁷³. In TYPOLIGHT dagegen erfolgt die Organisation von Dokumenten im DATEIMANAGER. Innerhalb des DATEIMANAGERS können verschiedene Verzeichnisstrukturen in hierarchische Anordnung erstellt werden. Dies ermöglicht das Ordnen und Kategorisieren von Dokumenten²⁷⁴. In WordPress hingegen können keine expliziten Verzeichnisse für die MEDIATHEK manuell angelegt werden, sodass die Sortierung der Dateien nach Erstellungsjahr bzw. Erstellungsmonat des Uploads erfolgt.²⁷⁵

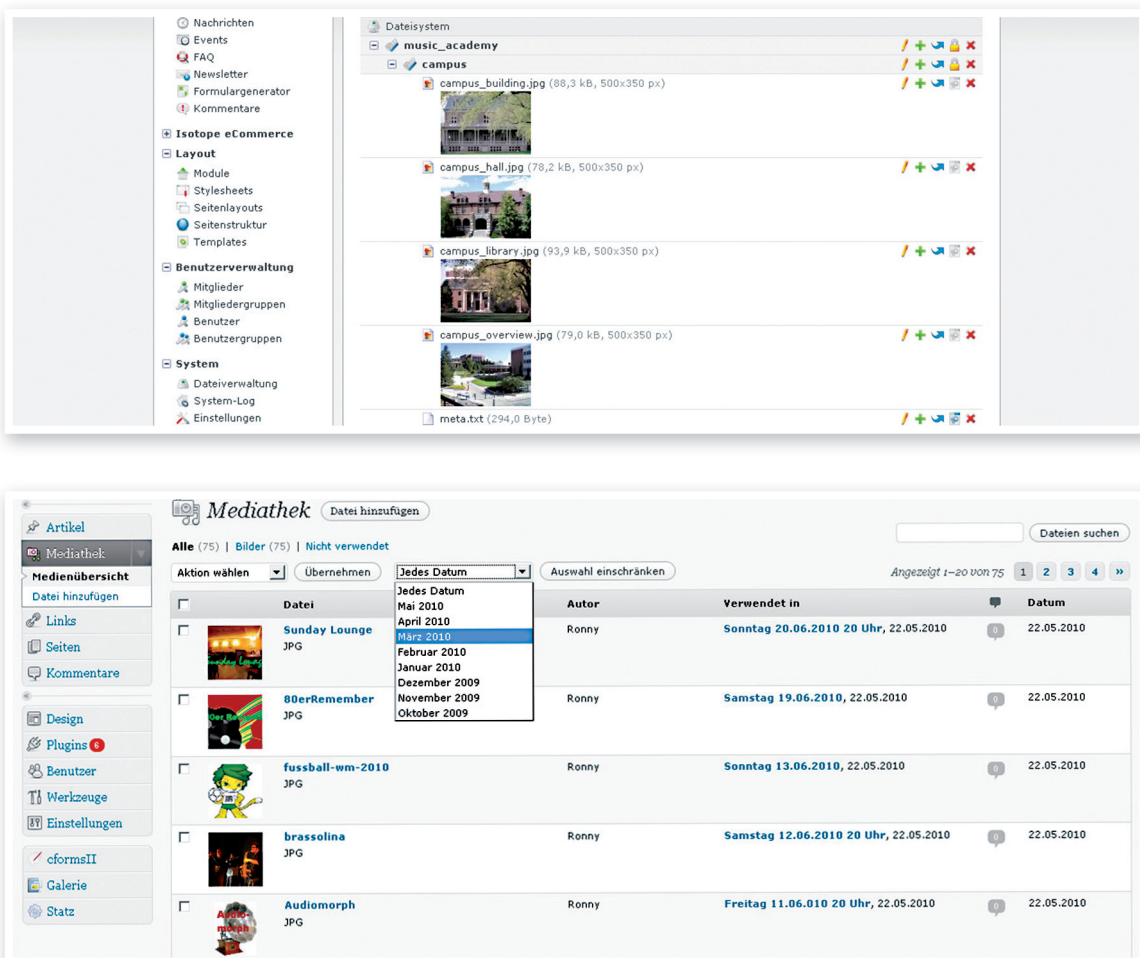


Abbildung 7.9: DATEIVERWALTUNG VON TYPOLIGHT SOWIE MEDIATHEK VON WORDPRESS

273 Siehe: Kapitel 4.2.3. Mediathek

274 Vgl. Feyer, 2010, S. 123

275 Vgl. Belik, 2010, S. 104ff.

Bei der Einbindung von Medieninhalten gibt es einen größeren Unterschied im Vergleich beider Systeme. So können in TYPOLIGHT, Medieninhalte in ARTIKEL oder SEITEN mit Hilfe von INHALTSELEMENTEN eingefügt oder direkt über den TINYMCE-EDITOR implementiert werden. Hierzu müssen die jeweiligen Medieninhalte bereits im Dateimanager hochgeladen und vorhanden sein. Ein Upload ist während des Erstellungsvorganges von SEITEN, ARTIKEL oder NACHRICHTEN nicht möglich. Eine komfortablere Vorgehensweise ermöglicht WordPress bei der Einbindung von Medien. Zwar können hier Medieninhalte nur über den TinyMCE-Editor eingebunden werden, dafür lassen sich diese auch während des Erstellungsvorgangs in das System integrieren. Gerade im Hinblick auf ‚Workflow‘ sind dies geringe Defizite von TYPOLIGHT gegenüber WordPress.

Die Einbindung von mehreren Bilddokumenten stellt sich in TYPOLIGHT eher als umständlich heraus. So werden innerhalb der TINYMCE-Umgebung von TYPOLIGHT keine Vorschaubilder sondern ausschließlich Dateinamen angezeigt. Gerade in der Praxis stellt sich diese Art der Einbettung vor allem für ungeübte Nutzer als schwierig und nicht benutzerfreundlich heraus.

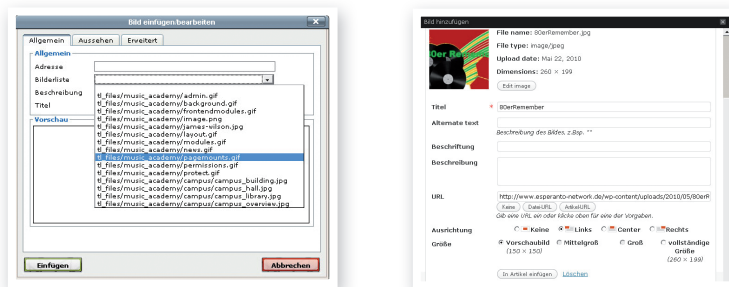


Abbildung 7.10: Einbinden von Bildern in TYPOLIGHT und WordPress

7.4 Rechteverwaltung

Die Rechteverwaltung ist eines der zentralen Themen eines WCMS. Die Grundfunktionalität der Rechteverwaltung ist in WordPress bereits im Kern integriert, sodass eine allgemeine Einschränkung über Benutzer getroffen wird²⁷⁶. Mittels PLUGINS lassen sich bestimmte Eigenschaften erweitern und ausbauen. Jedoch verdeutlichen praktische Erfahrungen, dass viele Segmente auch mit PLUGINS in WordPress nicht ausreichend abzudecken sind, wie zum Beispiel die Einrichtung von Frondend-Benutzern.

276 Vgl. hierzu Kapitel 4.2.9

TYPOLIGHT dagegen bietet eine große Anzahl an vorhandenen Kernfunktionen für die Verwaltung und Bearbeitung von Nutzern. So unterscheidet das System generell zwischen MITGLIEDERN (Frontend-Benutzer) und BENUTZER (Backend-Benutzer)²⁷⁷. Innerhalb dieser Benutzertypen lassen sich spezielle Gruppen mit unterschiedlichsten Eigenschaften definieren.

Der wesentliche Unterschied beider Systeme ist, dass in TYPOLIGHT, Rechte den BENUTZERN zugewiesen werden müssen. Grundsätzlich haben BENUTZER keinerlei Rechte und müssen erst vom Administrator bestimmt werden. Im Gegensatz zu WORDPRESS gibt es keine Vorkonfigurationen. Alle ROLLEN/BENUTZER darf ausschließlich der Administrator einrichten, welcher grundsätzlich alle Rechte besitzt.

Der Vorteil dieser TYPOLIGHT-Verwaltungsmethode ist vor allem die Genauigkeit. So können nicht nur die Zugriffe auf bestimmte Backend-Module eingeschränkt, sondern auch einzelne Eingabefelder unkenntlich gemacht werden. Damit hat der Systemadministrator die vollständige Kontrolle über das WCMS und ermöglicht unerfahrenen Nutzern eine minimale Benutzeroberfläche²⁷⁸. Im Vergleich zu WORDPRESS, ändert sich das Backend bei TYPOLIGHT mit eingeschränkten Nutzungsrechten erheblich.

Eine weitere Besonderheit in der Benutzerverwaltung von TYPOLIGHT ist die Gruppierung von BENUTZERN und MITGLIEDERN. Jeder BENUTZER bzw. jedes MITGLIED kann Angehöriger in mehreren BENUTZERGRUPPEN / MITGLEIDSGRUPPEN sein und erbt alle Rechte der zugewiesenen Gruppe. Die verschiedenen Berechtigungen werden addiert, sodass zum Beispiel ein BENUTZER der Gruppe A und der Gruppe B, die Summe der Rechte aus beiden Gruppen erhält.²⁷⁹

Die flexible und übersichtliche Benutzerverwaltung von TYPOLIGHT ergibt eine bestmögliche Schnittstelle zwischen Nutzer und System. Vor allem die Zuweisung bestimmter, personalisierter Eigenschaften macht diese Verwaltung bedeutsam. In WORDPRESS gelingt eine derartige Schnittstelle mit aktuellen PLUGINS nur sehr gering.

²⁷⁷ Vgl. Feyer, 2010, S. 234 f.

²⁷⁸ Vgl. Feyer, 2010, S. 234

²⁷⁹ Vgl. Feyer, 2010, S. 235

8. Resümee

Die theoretischen und praktischen Analysen dieser Arbeit zeigen, dass die Nutzung der Blogsoftware `WordPress` als Web Content Management System für die dargestellten Inhaltsverwaltungsszenarien geeignet ist und beachtliche Schnittmengen zwischen Weblog-Systemen und WCMS existieren. Gerade die intuitiv bedienbare und gut strukturierte Administrationsoberfläche von `WordPress` ermöglicht ein unkompliziertes Contenthandling und erlaubt dem Nutzer, ein schnelles Veröffentlichen von Inhalten. Darüber hinaus können vor allem klein- und mittelständige Unternehmen, sowie Privatpersonen von `WordPress` als WCMS profitieren, da die Blogsoftware eine Vielzahl an Grundfunktionen für die Nutzung als WCMS bereitstellt²⁸⁰. Eine unkomplizierte Installation, viele verschiedene `LAYOUTTHEMES` und eine stetig wachsende Entwicklergemeinschaft sorgen für einen schnellen Einstieg.

Die historische Entwicklung von `WordPress` verdeutlicht das Wachstum des Systems und schildert die Integration neuer, WCMS relevanter, `Features`²⁸¹. So sind neben der immer weiterentwickelten Administrationsoberfläche, die Implementierung von `SEITEN` sowie die Erweiterbarkeit des Systems durch `PLUGINS` und `THEMES` hervorzuheben, welche die generelle Nutzung von `WordPress` als WCMS ermöglichen. Aber auch die Verwaltung von Benutzerrollen und die Vergabe von Rechten zählen zu den Kernpunkten für einen optimalen Einsatz von `WordPress` als WCMS. Wie allerdings praktische Analysen bzw. Erfahrungen ergaben, besteht hierbei im `CORE` des Systems Optimierungsbedarf, um `WordPress` mit detaillierten und benutzerbezogenen Rechten auszustatten. Dieses vermeintliche Defizit lässt sich zum heutigen Zeitpunkt nicht oder nur ungenügend durch `PLUGINS` kompensieren, welche zu den elementarsten Funktionen von `WordPress` innerhalb eines WCMS gehören.

`PLUGINS` erweitern `WordPress` um zahlreiche Funktionen und ermöglichen damit den Einsatz als WCMS. So können beispielsweise Formulare, Galerien, Gästebücher, Umfragen etc. in das System eingebunden werden, welche standardmäßig nicht im `CORE` vorhanden sind. `PLUGINS` werden in der Regel von externen Entwicklern programmiert und frei zum Download zur Verfügung gestellt. Hierbei besteht jedoch die Gefahr, dass Sicherheitslücken in das System gelangen und diese Beschädigungen innerhalb des Systems verursachen. Daher ist der Einsatz von `PLUGINS` vor dem Gebrauch auf Sicherheitsstandards und Kompatibilität zu prüfen und je nach Ergebnis abzuwägen – vor allem Erfahrungsberichte von Nutzern können dabei hilfreich sein. Einen Überblick aller offiziell zugelassenen Erweiterungen erhält man auf der Betreiberplattform ‚www.wordpress.org‘.

280 Vgl. hierzu Kapitel 4

281 Vgl. hierzu Kapitel 3

Viele der im Kapitel 2 definierten Anforderungen, die an ein Web Content Management System gestellt werden, erweisen sich für WordPress als belegt²⁸². Herausragend ist dabei vor allem die einfache Bedienbarkeit des Backends, die Gewährleistung eines einheitlichen Designs durch Themes, die Integrationsfähigkeiten und Anpassungen durch Plugins sowie überschaubare Kosten und die Sicherstellung von Industriestandards bzw. Barrierefreiheit. Schwächen hingegen werden in Sicherheitsaspekten und in der Mehrsprachigkeit im WordPress-Funktionsumfang nachgewiesen, die sich jedoch durch Erweiterungen beheben lassen.

Aber auch Systemfeatures wie der WYSIWYG-Editor, XML-Schnittstellen, Feeds, Multiuser, zeitgesteuerte Publizierung, Archiv- und Kommentarfunktion beinhaltet das System oder können durch entsprechende Erweiterungen nachgerüstet werden. Im Verlauf der Arbeit ergab sich jedoch auch, dass einige Funktionen die WordPress im Core beinhaltet, für die Nutzung als WCMS unnötig sind, wie beispielsweise die Verwendung von Trackbacks oder Pingbacks. Auch die Differenzierung in geschützte Inhaltsbereiche für Nutzer gestaltet sich mit WordPress schwierig, wie der direkte Vergleich mit dem WCMS TypoLight belegt.²⁸³

Ebenfalls werden bei der Implementierung eines Terminkalenders und der Darstellung von Terminen in WordPress Grenzen deutlich, die nur durch ergänzende Funktionen wie separate Plugin-Programmierung oder anpassen von Templates sich realisieren lassen²⁸⁴. Des Weiteren kann bei einem großen Inhaltsumfang innerhalb der Seiten und Artikel, die Übersichtlichkeit über bereits getroffene Formatierungen verloren gehen, da WordPress immer nur eine Sektion für die Inhaltseingabe bereitstellt. Eine Aufteilung in mehrere Inhaltselemente wie bei TypoLight würde dieser Unübersichtlichkeit mindern. Dennoch zeigt sich im direkten Vergleich zu TypoLight²⁸⁵, dass WordPress durchaus im Stande ist, sich in die Reihe der marktführenden Web Content Management Systemen im Bereich Open Source einzugliedern und nicht ausschließlich als Blogsoftware anzusehen ist.

Für zukünftige Versionen wäre es sinnvoll, bei immer wachsender Beliebtheit des Systems, sich verstärkt auf sicherheitstechnische Aspekte zu konzentrieren und eventuell Sicherheitslücken zu schließen. Auch die Implementierung der für die Version 3.0 angekündigten ‚Custom Post Types‘ und ‚Custom Taxonomies‘ würden den WCMS-Funktionsumfang von WordPress positiv ergänzen.

Festzuhalten ist, dass für klein- und mittelgroße Projekte, der Einsatz von WordPress als WCMS sich gerade für unerfahrene Nutzer eignet und Projekte bei geringem Budget sich schnell realisieren lassen.

282 Vgl. hierzu Kapitel 2

283 Vgl. hierzu Kapitel 7

284 Vgl. hierzu Kapitel 6

285 Vgl. hierzu Kapitel 7

IV. Anhang

CONDITIONAL TAGS

is_404(): Es wird abgefragt, ob es sich um eine Fehlerseite handelt.

is_category(): Es wird die Kategorienseite ausgegeben. Dieser TAG kann Parameter enthalten um gezielte Kategorien mit dem jeweiligen Namen, den Titel, der ID auszuliefern.

in_category(): Es wird angezeigt, ob der aktuelle Beitrag sich in einer bestimmten Kategorie befindet. Dieser TAG kann Parameter enthalten.

is_page(): Es wird abgefragt, ob es sich um eine WORDPRESS-SEITE handelt. Dieser TAG kann Parameter wie ID, Titel und eine Titelform beinhalten.

is_home(): Es wird abgefragt, ob es sich um die Startseite von WORDPRESS handelt.

is_single(): Es wird abgefragt, ob es sich um einen einzelnen Beitrag handelt. Dieser TAG kann ebenfalls Parameter wie Namen, Titel, Titelform und ID beinhalten.

is_single(10'): Ausgabe des Beitrags mit der ID ,10'.

is_single(Portfolio'): Ausgabe des Beitrags mit dem Titel ,Portfolio'.

is_single(Hallo Welt'): Ausgabe des Beitrags mit der Titelform „Hallo Welt“.

is_single(array(10, 12, 16)): Ausgabe des Beitrags wenn innerhalb des definierten ,arrays' entweder die ID 10, 12 oder 16 aufgerufen werden. Die Array-Fähigkeit innerhalb von CONDITIONAL TAGS besteht seit der WORDPRESS Version 2.5. Diese Funktion differenziert dabei nicht zwischen der ID, dem Titel oder dem Name.

is_search(): Es wird abgefragt, ob es sich um das Ergebnis der Suchabfrage handelt.

is_archive(): Es wird abgefragt, ob es sich um die Archivseite handelt. Dabei ist gleichgültig auf welcher Kategorie, Autor oder Zeitpunkt die SEITE basiert.

is_date(): Es wird abgefragt, ob es sich um eine datumbasierte Archivseite handelt. Dieser Tag verwendet keine Parameter.

is_year(): Es wird abgefragt, ob es sich um eine Jahresarchivseite handelt. Auch dieser Tag kann keine Parameter enthalten.

is_month(): Es wird abgefragt, ob es sich um eine Monatsarchivseite handelt. Ebenfalls keine Verwendung von Parameter.

is_day(): Es wird abgefragt, ob es sich um eine Tagesarchivseite handelt. Ebenfalls keine Verwendung von Parameter.

is_time(): Es wird abgefragt, ob es sich um eine Stunden-, Minuten-, und Sekundenarchivseite handelt.

V. Literaturverzeichnis

Adler, Olivia (2009): Praxiswissen Wordpress
Köln: O'Reilly Verlag

Alby, Tom (2008a): Professionell Bloggen mit Wordpress
München: Carl Hanser Verlag

Alby, Tom (2008b): Web 2.0: Konzepte, Anwendungen, Technologien
3., überarbeitete Auflage. München: Carl Hanser Verlag

Alby, Tom & Karzauninkat, Stefan (2007): Suchmaschinenoptimierung:
Professionelles Website-marketing für besseres Ranking,
2. aktualisierte Auflage, München: Carl Hanser Verlag

Abold, Roland (2005): Wahlkampf in der Blogosphäre
[www-dokument] URL: http://www.uni-bamberg.de/kommunikation/news/archiv_und_suche/wintersemester_20052006/artikel/wahlkampf/
Stand: 17.04.2010

Belik, Jolantha (2010): WordPress 2.8 - Vom einfachen Blog zur dynamischen Website
München: Addison Wesley Verlag.

Behrens, Kevin (2010): Role Scoper 1.2.8 Beta 2
URL: <http://agapetry.net/news/role-scoper-128-beta-2/>
Stand: 17.04.2010

Burnette, Ed (2007): Movable Type becomes free again
[www-dokument] URL: <http://blogs.zdnet.com/Burnette/?p=328>
Stand: 18.04.2010

Bültge, Frank (2007): WordPress. Weblogs einrichten und administrieren
München: open source press

Bültge & Boley (2009): Das WordPress Buch. Vom Blog zum Content-Management-System.
München: open source press

Bültge, Frank (2009): WordPress Konstanten nutzen

[www-dokument] URL: <http://buelgtge.de/wordpress-konstanten-nutzen/962/>

Stand: 05.Mai 2010

Bültge, Frank (2008): Sticky-Funktionalität ab WordPress 2.7

[www-dokument] URL: <http://buelgtge.de/sticky-funktionalitaet-wordpress-27/788/>

Stand: 05.Mai 2010

Bültge, Frank (2006): WP - Gästebuch [WP-Guestbook] (Plugin)

[www-dokument] URL: <http://buelgtge.de/wp-gaestebuch-wp-guestbook-plugin/155/>

Stand: 05.Mai 2010

Bültge, Frank (2008): WordPress Login Sicherheit - Secure WP

[www-dokument] URL: <http://buelgtge.de/wordpress-login-sicherheit-plugin/652/>

Stand: 05.Mai 2010

Bültge, Frank (2007): WP - Tutorial, wir schreiben ein WordPress Theme - Teil 3

[www-dokument] URL: <http://buelgtge.de/wp-tutorial-wir-schreiben-ein-wordpress-theme-teil-3/527/> Stand: 15.05.2010

Cannaday; Key & Kuehl (2009): Using sIFR in Wordpress

[www-dokument] URL: <http://www.switchonthecode.com/tutorials/using-sifr-in-wordpress>

Stand: 15.05.2010

Chan Hong Wie, Lester (2010): WP-Polls 2.50

[www-dokument] URL: <http://lesterchan.net/portfolio/programming/php/#wp-polls>

Stand: 15.05.2010

Curly, Rob (2007): NextGen-Gallery

[www-dokument] URL: http://www.curlyrob.de/curlyrob/?page_id=129

Stand: 15.05.2010

Eenfeldt, Johan (2009): Limit Login Attempts

[www-dokument] URL: <http://devel.kostdoktorn.se/limit-login-attempts>

Stand: 02.06.2010

Feyer, Leo (2010): Typolight Open Source CMS Funktionen

[www-dokument] URL: <http://www.typolight.org/funktionen.html>

Stand: 05.Mai 2010

Feyer, Leo (2010): Typolight Open Source CMS Funktionen
[www-dokument] URL: <http://www.contao.org/neuigkeiten/items/typolight-wird-zu-contao.html#new-name> Stand: 02.06.2010

Feyer, Leo (2010): Das offizielle Typolight-Handbuch
2.Auflage. München: Pearson EducationDeutschland GmbH

Fitton, Laura; Gruen Michael & Poston Leslie (2009): Twitter For Dummies
Hoboken: Wiley Publishing

Frütel, Thomas (2009): Das große Buch Wordpress professionell einsetzen
Düsseldorf: Data Becker

Lobbo, Sachsa (2009): Morgendämmerung der Bloglandschaft
[www-dokument] URL: <http://saschalobo.com/2009/02/04/morgendaemmerung-der-bloglandschaft/>, Stand: 18.04.2010

Jerkovic, John (2010): SEO Warrior
2. Auflage. Sebastopol: O Reilly Media

Graf, Hagen (2008): Joomla! 1.5: Websites organisieren und gestalten mit dem Open Source-CMS. 10. Auflage. München: Addison Wesley Verlag

Grimme Online Award (2005): Adolf Grimme Institut. Grimme Online Award. Marl.
[www-dokument] URL: <http://www.grimme-institut.de/html/index.php?id=163>
Stand: 12. Februar 2010

Heilemann, Michael (2010): Kubrick (DE-Edition) 1.6
[www-dokument] URL: <http://themes.wordpress-deutschland.org/kubrick-de-edition/>
Stand: 12. Mai 2010

Herring; Kouper; Paolillo; Scheidt; Tyworth; WelschWright & Yu (2005): Conversations in the Blogosphere: An Analysis "From the Bottom Up". Bloomington: Indiana University Bloomington.
[www-dokument] URL: <http://ivl.slis.indiana.edu/km/pub/2005-herring-blogosph.pdf>
Stand: 12.02.2010

Huesmann, Theo (2008): Wordpress Revision Management wirkungsvoll nutzen
[www-dokument] URL: <http://gumia.de/wordpress-revision-management/>
Stand: 12.Mai 2010

Kantel, Jörg (2003): Vom Weblog lernen... Community, Peer-to-Peer und Eigenständigkeit als Modell für zukünftige Wissenssammlungen.

[www-dokument] URL: <http://echo.mpiwg-berlin.mpg.de/home/publicRelation/talks/images/kantel20030522.pdf> Stand: 12. Februar 2010

Kemper, Mehanna; Unger (2006): Business Intelligence. Grundlagen und praktische Anwendungen. Eine Einführung in die IT-basierte Managementunterstützung.

2. ergänzte Auflage. Wiesbaden: Vieweg & Sohn Verlag

Koerber, Susanne (2009): Vor der Wahl: Parteien von Twitter völlig überfordert

[www-dokument] URL: <http://www.pr-com.de/de/pressezentrum/presseinformationen/index.php?ID=7c0525611eb15bcc862f9b1eae4ab377>

Stand: 18.04.2010

Klätke, Tom (2008): NetzNews. [www-dokument] URL: <http://www.netznews.org/?p=1034>

Stand: 12.02.2010

Lindner, Stefanie (2009): Die Beobachtung der Beobachter: monothematische Medienkritik an Leitmedien. Hamburg: Academic Transfer

Lotze & Theune (2005): Content Management mit Plone. Handbuch für Autoren und Redakteure
Köthen: gocept GmbH & Co

Löwer, Chris (2006): Digitale Mundpropaganda. Zeit-Online. Hamburg.

[www-dokument] URL: <http://www.zeit.de/2006/30/Blogs>

Stand: 12.02.2010

McCullagh & Broache (2007): Blogs turn 10 - who's the father?. Someone, somewhere created the very first Web log. It's just not quite clear who.

[www-dokument] URL: http://news.cnet.com/2100-1025_3-6168681.html

Stand: 17.04.2010

McGiboney, Michelle (2009): Twitter's Tweet Smell Of Success

[www-dokument] http://blog.nielsen.com/nielsenwire/online_mobile/twitters-tweet-smell-of-success/ Stand: 12.02.2010

Metzmacher, Jan (2010): Einblick in das CMS „TYPOlight“

[www-dokument] URL: <http://www.drweb.de/magazin/einblick-in-das-cms-typolight/>

Stand: 02.06.2010

Meyer-Gellersen, Lars (2006): HowTo: Wordpress Widgets richtig nutzen
[www-dokument] URL: <http://bloganbieter.de/2006/06/09/wordpress-widgets-richtig-nutzen/>
Stand: 02.06.2010

Mullenweg, Matt (2010): Roadmap
[www-dokument] URL: <http://wordpress.org/about/roadmap/>
Stand: 18.04.2010

Mullenweg, Matt (2004): WordPress 1.0
[www-dokument] URL: <http://wordpress.org/development/2004/01/wordpress-10/>
Stand: 18.04.2010

Mullenweg, Matt (2004): 1.0.1 Release Candidate
[www-dokument] URL: <http://wordpress.org/development/2004/01/101-release-candidate/>
Stand: 18.04.2010

Mullenweg, Matt (2004): Changelog/1.0.2
[www-dokument] URL: <http://codex.wordpress.org/Changelog/1.0.2>
Stand: 18.04.2010

Mullenweg, Matt (2004): 1.2 Release Candidate
[www-dokument] URL: <http://wordpress.org/development/2004/05/12-release-candidate/>
Stand: 18.04.2010

Mullenweg, Matt (2005): Announcing WordPress 1.5
[www-dokument] URL: <http://wordpress.org/development/2005/02/strayhorn/>
Stand: 18.04.2010

Mullenweg, Matt (2005): Wordpress 2.0
[www-dokument] URL: <http://wordpress.org/development/2005/12/wp2/>
Stand: 18.04.2010

Mullenweg, Matt (2007): WordPress 2.1 Ella
[www-dokument] URL: <http://wordpress.org/development/2007/01/ella-21/>
Stand: 18.04.2010

Mullenweg, Matt (2007): WordPress 2.2
[www-dokument] URL: <http://wordpress.org/development/2007/05/wordpress-22/>
Stand: 23.04.2010

Mullenweg, Matt (2007): WordPress 2.3

[www-dokument] URL: <http://wordpress.org/development/2007/09/wordpress-23/>

Stand: 23.04.2010

Mullenweg, Matt (2008): WordPress 2.5

[www-dokument] URL: <http://wordpress.org/development/2008/03/wordpress-25-brecker/>

Stand: 23.04.2010

Mullenweg, Matt (2008): WordPress 2.6

[www-dokument] URL: <http://wordpress.org/development/2008/07/wordpress-26-tyner/>

Stand: 23.04.2010

Mullenweg, Matt (2008): WordPress 2.7

[www-dokument] URL: <http://wordpress.org/development/2008/12/coltrane/>

Stand: 23.04.2010

Mullenweg, Matt (2009): 2.8 Release Jazzes Themes and Widgets

[www-dokument] URL: <http://wordpress.org/development/2009/06/wordpress-28/>

Stand: 23.04.2010

Mullenweg, Matt (2010): Requirements

[www-dokument] URL: <http://wordpress.org/about/requirements/>

Stand: 05.05.2010

Mullenweg, Matt (2010): Installing WordPress

[www-dokument] URL: http://codex.wordpress.org/Installing_WordPress

Stand: 05.05.2010

Mullenweg, Matt (2010): Administration Panels

[www-dokument] URL: http://codex.wordpress.org/Administration_Panels#Dashboard

Stand: 05.05.2010

Mullenweg, Matt (2010): Dashboard Updates SubPanel

[www-dokument] URL: http://codex.wordpress.org/Dashboard_Updates_SubPanel.

Stand: 05.05.2010

Mullenweg, Matt (2010): Template Tags/the title

[www-dokument] URL: http://codex.wordpress.org/Template_Tags/the_title.

Stand: 05.05.2010

Mullenweg, Matt (2010): Template Tags/the excerpt

[www-dokument] URL: http://doku.wordpress-deutschland.org/Template_Tags/the_excerpt.

Stand: 05.05.2010

Mullenweg, Matt (2010): Custom_Fields

[www-dokument] URL: http://codex.wordpress.org/Custom_Fields

Stand: 05.05.2010

Mullenweg, Matt (2010): Pages

[www-dokument] URL: <http://codex.wordpress.org/Pages>

Stand: 05.05.2010

Mullenweg, Matt (2010): Changing File Permissions

[www-dokument] URL: http://codex.wordpress.org/Changing_File_Permissions

Stand: 15.05.2010

Mullenweg, Matt (2010): Users Your Profile SubPanel

[www-dokument] URL: http://codex.wordpress.org/Your_Profile_SubPanel

Stand: 15.05.2010

Mullenweg, Matt (2010): Updating WordPress

[www-dokument] URL: http://codex.wordpress.org/Upgrading_WordPress

Stand: 15.05.2010

Mullenweg, Matt (2010): Using Permalinks

[www-dokument] URL: http://codex.wordpress.org/Using_Permalinks

Stand: 15.05.2010

Mullenweg, Matt (2010): Include Tags

[www-dokument] URL: http://codex.wordpress.org/Include_Tags

Stand: 15.05.2010

Mullenweg, Matt (2010): The Loop

[www-dokument] URL: http://codex.wordpress.org/The_Loop

Stand: 15.05.2010

Mullenweg, Matt (2010): Template Tags

[www-dokument] URL: http://codex.wordpress.org/Template_Tags

Stand: 15.05.2010

Mullenweg, Matt (2010): How to Pass Tag Parameters

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/How_to_Pass_Tag_Parameters. Stand: 15.05.2010

Mullenweg, Matt (2010): How to Pass Tag Parameters / Types of parameters

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/How_to_Pass_Tag_Parameters#Types_of_parameters. Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/bloginfo

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/bloginfo Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/list_cats

[www-dokument] URL:http://codex.wordpress.org/Function_Reference/list_cats. Stand: 15.05.2010

Mullenweg, Matt (2010): Conditional Tags

[www-dokument] URL:http://codex.wordpress.org/Conditional_Tags. Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/query_posts

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/query_posts. Stand: 15.05.2010

Mullenweg, Matt (2010): Creating Your Own Page Templates

[www-dokument] URL:http://codex.wordpress.org/Pages#Creating_Your_Own_Page_Templates Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/wp_list_pages

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/wp_list_pages Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/wp_list_pages Default Usage

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/wp_list_pages#Default_Usage . Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/wp_list_pages Parameters

[www-dokument] URL:http://codex.wordpress.org/Template_Tags/wp_list_pages#Parameters Stand: 15.05.2010

Mullenweg, Matt (2010): Function Reference/wp enqueue script
[www-dokument] URL:http://codex.wordpress.org/Function_Reference/wp_enqueue_script
Stand: 15.05.2010

Maurice, Florence (2007): Web 2.0-Praxis
München: Pearson Education

Nix; Handke; Kampffmeyer; Krüger; Wechner & Zschau (2005): Web Content Management.
CMS verstehen und auswählen.
Frankfurt: Software & Support Verlag

Petereit, Dieter (2009): Wordpress-Sites in mehreren Sprachen
[www-dokument] URL:<http://www.drweb.de/magazin/wordpress-sites-in-mehreren-sprachen/>
Stand: 15.05.2010

Puscher, Frank (2005): SIFR: Neue Schriftenvielfalt
[www-dokument] URL:<http://www.drweb.de/magazin/sifr-neue-schriftenvielfalt/>.
Stand: 15.05.2010

OnTheGoSystems, Inc. (2010) Getting Started Guide
[www-dokument] URL: <http://wpml.org/documentation/getting-started-guide/>
Stand: 15.05.2010

o.A. (2008): Core. Wordpress Deutschland [www-dokument] URL:<http://forum.wordpress-deutschland.org/lexikon/core-52/alle-eintraege-1.html>
Stand: 05.Mai 2010

o.A. (2009): Page Attributes
[www-dokument] URL:<http://en.support.wordpress.com/pages/page-attributes/>
Stand: 05.Mai 2010

o.A. (2009): Was ist ein Theme?
[www-dokument] URL:http://doku.wordpress-deutschland.org/Themes_benutzen#Was_ist_ein_Theme.3F
Stand: 05.Mai 2010

o.A. (2010): WordPress Plug-ins ?
[www-dokument] URL:<http://doku.wordpress-deutschland.org/Plug-in>
Stand: 15.Mai 2010

o.A. (2010): Plug-in Übersicht

[www-dokument] URL:http://doku.wordpress-deutschland.org/Plugin_Management

Stand: 15.05.2010

o.A. (2010): Allgemeine Einstellungen

[www-dokument] URL:http://doku.wordpress-deutschland.org/Allgemeine_Einstellungen:

15.05.2010

o.A. (2010): date

[www-dokument] URL:<http://php.net/manual/de/function.date.php>

Stand: 15.05.2010

o.A. (2010): Funktionsparameter

[www-dokument] URL:<http://de2.php.net/manual/de/functions.arguments.php>

Stand: 15.05.2010

Oeser, Michael (2009): Wie man WordPress Textauszüge mit the_excerpt individuell anpasst

[www-dokument] URL: http://www.michaeloeser.de/wie-man-wordpress-textauszuege-mit-the_excerpt-individuell-anpasst-24700.html

Stand: 05.05.2010

Onasch, Lars (2006): Spezielle Anforderungen an Content Management Systeme für den Mittelstand.

[www-dokument] URL: http://www.contentmanager.de/magazin/artikel_917-print_content_management_mittelstand_auswahl.html

Stand: 13. Februar 2010

Ozz, Andrew (2009): TinyMCE Advanced

[www-dokument] URL: <http://www.laptoptips.ca/projects/tinymce-advanced/>

Stand: 05.05.2010

Presserat (2008): Deutscher Presserat. Pressekodex. Berlin. [www-dokument] URL:

<http://www.presserat.info/Pressekodex.pressekodex.0.html>

Stand: 12.02.2010

Piringer, Birgit (2009): Tipps und Tricks für Adult-Webmaster

Hamburg: Tredition GmbH

Rabe, Alex (2010) : NextGEN Gallery

[www-dokument] URL: <http://alexrabe.de/wordpress-plugins/nextgen-gallery/>

Stand: 15.05.2010

Sauer, Moritz (2007): Weblogs, Podcasting und Online-Journalismus.
Köln: O'Reilly Verlag

Siefken, Jan (2008): Internetmarketing durch Suchmaschinenoptimierung von Webseiten:
Mehr Besucher durch Suchmaschinen.
Nordstedt: Books on Demand GmbH

Schmitz, Olaf (2009): WP und WPMU sollen vereint werden
[www-dokument] URL: <http://blog.wordpress-deutschland.org/2009/05/30/wp-und-wpmu-sollen-vereint-werden.html>
Stand: 22.04.2010

Schmitz, Olaf (2009): WordPress 2.7 wurde veröffentlicht
[www-dokument] URL: <http://blog.wordpress-deutschland.org/2008/12/11/wordpress-27-wurde-veroeffentlicht.html#werkzeuge>
Stand: 15.05.2010

Schmitz, Olaf (2006): WordPress Widgets
[www-dokument] URL: <http://blog.wordpress-deutschland.org/2006/04/14/wordpress-widgets.html> . Stand: 15.05.2010

Schneider, Thomas (2007): Role Manager - What does Role Manager do?
[www-dokument] URL: <http://www.im-web-gefunden.de/wordpress-plugins/role-manager/>
Stand: 15. Mai 2010

Schröer, Lutz (2010): pagebar v2
[www-dokument] URL: <http://www.elektroelch.de/hacks/wp/pagebar/>
Stand: 15. Mai 2010

Simon, Nicole (2008): Deutsche Politiker und Parteien auf Twitter.
[www-dokument] URL: <http://mit140zeichen.de/deutsche-politiker-und-parteien-auf-twitter-114> . Stand: 18.04.2010

Simovic, Vladimir (2007): Das Einsteigerseminar Wordpress.
Kaarst. bhv Verlag

Simovic, Vladimir (2007): WordPress-Themes verstehen 2
[www-dokument] URL: <http://www.perun.net/2007/08/08/wordpress-themes-verstehen-2/>
Stand: 15. Mai 2010

Simoiv, Vladimir & Bonfranchi-Simovic, Thordis (2010): WordPress: das Praxisbuch
3. Auflage. Heidelberg: Hüthig Jehle Rehm

Sixtus, Mario (2005): Wachsame „Blogwarte“
[www-dokument] URL: http://www.fr-online.de/in_und_ausland/multimedia/blogosphaere/?em_cnt=651419&&sid=9d381de163852c0c1b3cadb57134f2ea
Stand: 17.04.2010

Sonnabend, Linda (2005): Deutsche Blogosphäre
[www-dokument] URL: <http://www.netzthemen.de/sonnabend-weblogs/2-2-4-deutsche-blogosphere> .
Stand: 17.04.2010

Stich, Nicole (2010): cformsII
[www-dokument] URL: <http://www.deliciousdays.com/cforms-plugin/>
Stand: 15.05.2010

Stöcker, Christan (2005): Blogger heizen Jamba ein
[www-dokument] URL: <http://www.spiegel.de/netzwelt/web/0,1518,335622,00.html>
Stand: 17.04.2010

Thon-Soun, Monika (2006): Trackback – Pingin- Tutorial
[www-dokument] URL: http://codex.wordpress.org/Template_Tags/the_title.
Stand: 05.05.2010

Trost, Silke (2005): Nielsen präsentiert seine Studie der neuen Medien: Deutsche Politiker wandeln auf den (Online-)Spuren von Barack Obama
[www-dokument] URL: <http://de.nielsen.com/news/NielsenPressemeldung11.02.09-Twitter.shtml> . Stand: 19.04.2010

Torbert, Michael (2010): All in One SEO Pack
[www-dokument] URL: <http://semperfiwebdesign.com/portfolio/wordpress/wordpress-plugins/all-in-one-seo-pack/>
Stand: 04.06.2010

Walter, Thomas (2008): Kompendium der Web-Programmierung. Dynamische Web-Sites.
Berlin, Heidelberg: Springer Verlag

Weber, Michael (2002): Web Content Management- Studie: Aktuelle Situation und Trends im Webpublishing.

[www-dokument]. URL: http://www.contentmanager.de/magazin/artikel_212_web_content_management_studie.html

Stand: 13.02.2010

Wetzlmayr, Robert (2008): Die neue Mediathek in WordPress 2.5

[www-dokument]. URL: <http://talkpress.de/artikel/medienbibliothek-wordpress-2-5>

Stand: 05.Mai 2010

Wöhrrer, Michael (2006): Software Guide. München.

[www-dokument] URL: <http://sw-guide.de/webdienste-blogging/was-sind-trackbacks/>

Stand: 12.Februar 2010

Zelend, Marcus (2008) : Wie kann ich mein Theme Widget-fähig machen?

[www-dokument] URL: <http://faq.wordpress-deutschland.org/wie-kann-ich-mein-theme-widget-faehig-machen/> Stand: 12.Mai 2010

Zerfaß & Bogosyan (2007): Blogstudie 2007. Informationssuche im Internet- Blogs als neues Recherchetool (Ergebnisbericht).Leipzig: Universität Leipzig.

[www-dokument] URL: http://www.blogstudie2007.de/inc/blogstudie2007_ergebnisbericht.pdf

Stand: 12. Februar 2010

VI. Erklärung zur selbstständigen Anfertigung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Alle Teile, die wörtlich oder sinngemäß einer Veröffentlichung entstammen, sind als solche kenntlich gemacht.

Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.

Datum: 27. August 2010

Unterschrift:



[Maik Irmscher]